



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

***PENGEMBANGAN SOFTWARE UNIT COMMITMENT AND
ECONOMIC DISPATCH DENGAN MEMPERTIMBANGKAN
RAMP-RATE***

Wildan Nurwanto
NRP 2211 100 167

Dosen Pembimbing
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
Ir. Arif Musthofa, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

**UNIT COMMITMENT AND ECONOMIC DISPATCH SOFTWARE
DEVELOPMENT CONSIDERING *RAMP-RATE***

Wildan Nurwanto
NRP 2211 100 167

Advisor
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
Ir. Arif Musthofa, M.T.

DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technologi
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**PENGEMBANGAN SOFTWARE UNIT COMMITMENT AND
ECONOMIC DISPATCH DENGAN MEMPERTIMBANGKAN
RAMP-RATE**

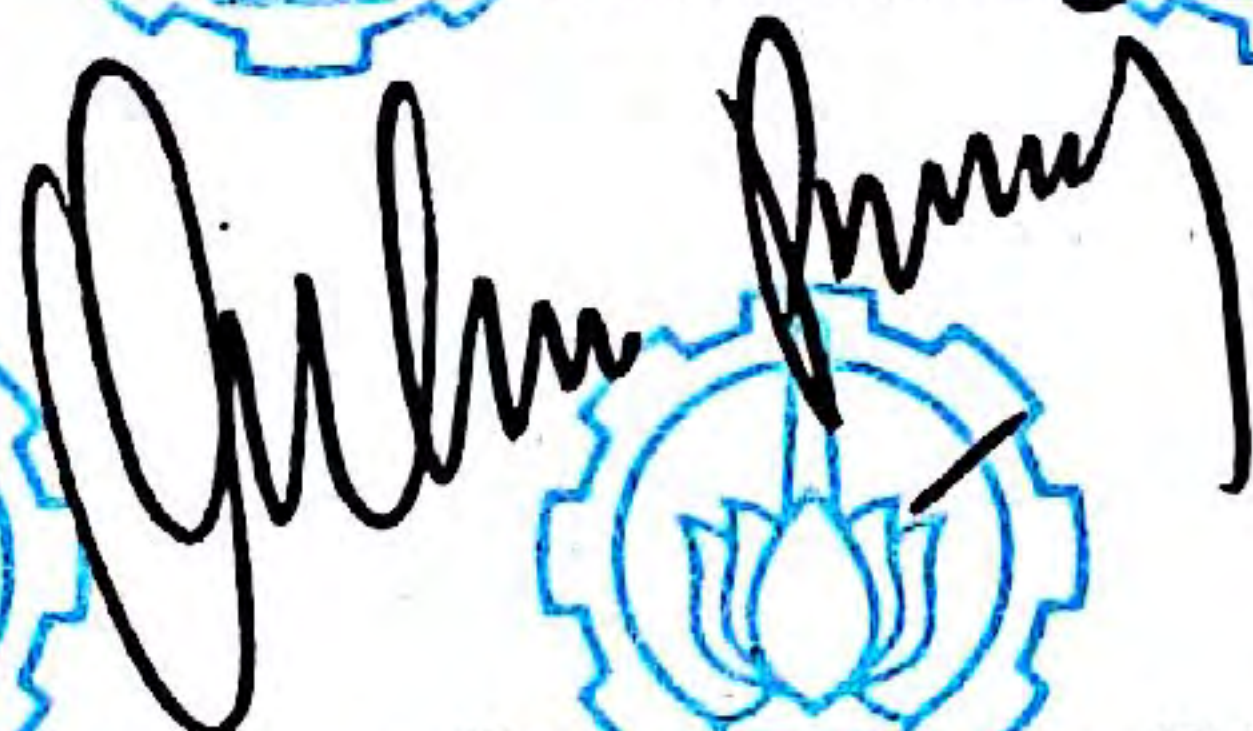
TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Teknik Sistem Tenaga
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

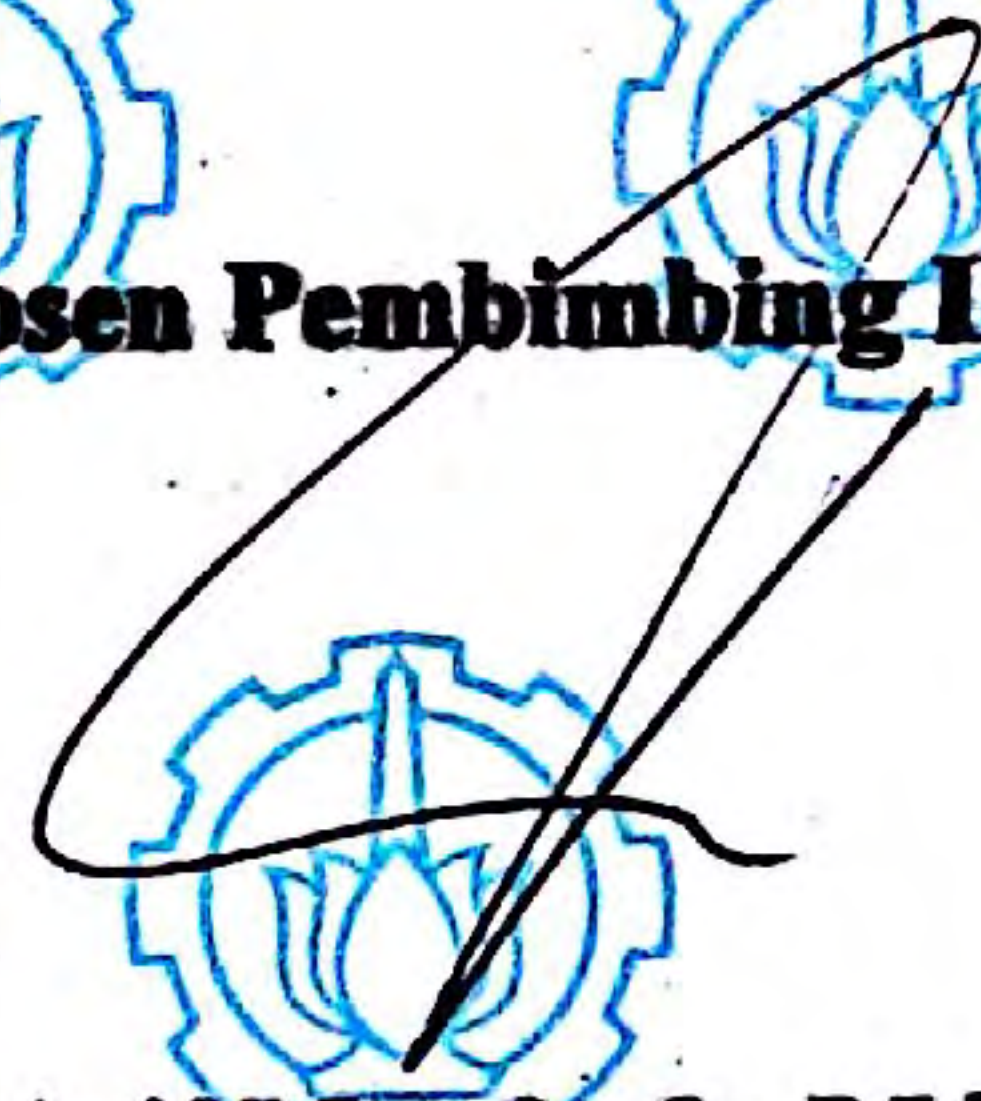
Menyetujui:

Dosen Pembimbing I

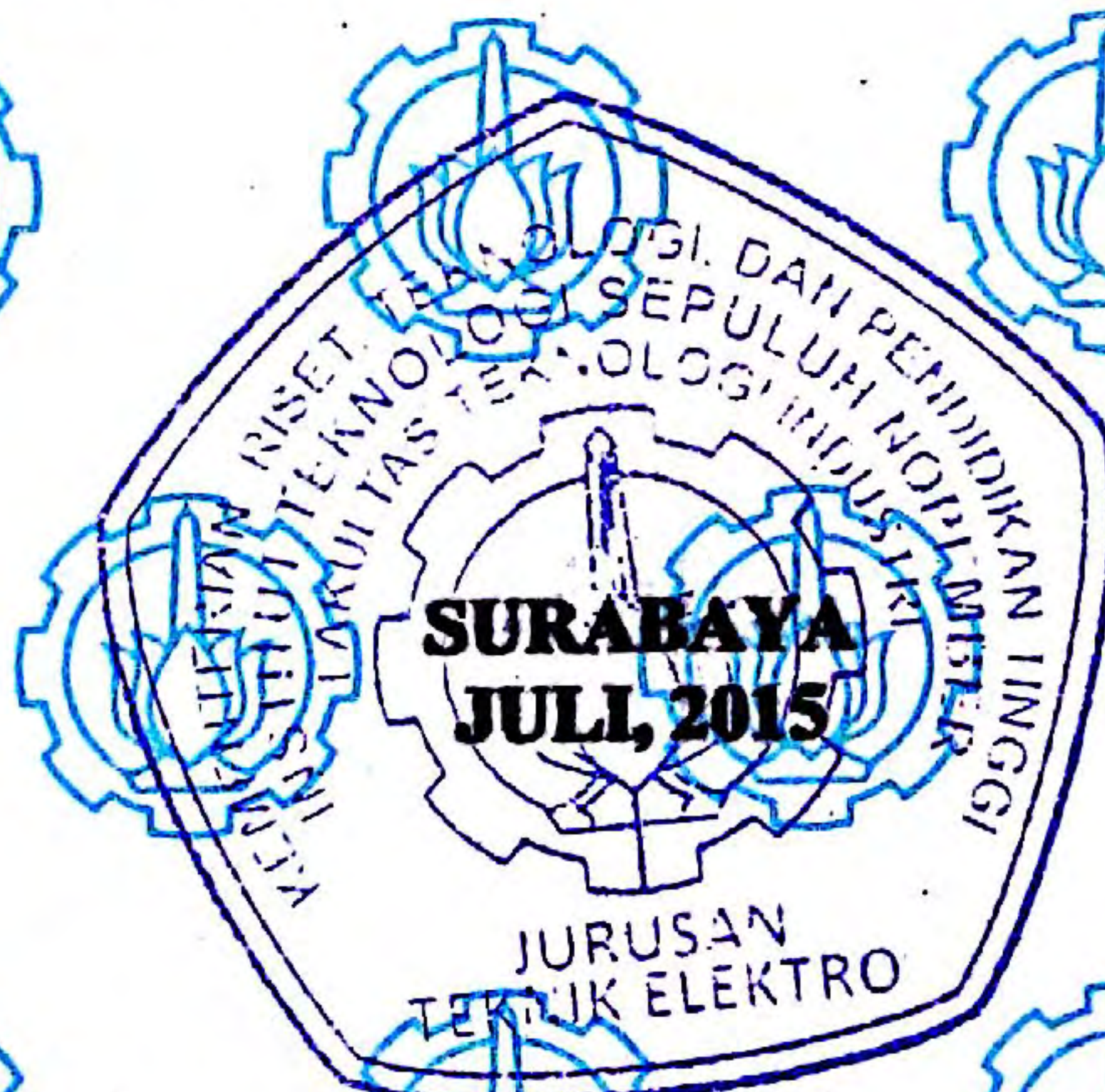


Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
NIP. 1949 07 15 1974 12 1001

Dosen Pembimbing II



Ir. Arif Musthofa, M.T.
NIP. 1966 08 11 1992 03 1004



Unit Commitment and Economic Dispatch Software Development Considering Ramp-rate

Name : Wildan Nurwanto
NRP : 2211100167
Advisor I : Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
Advisor II : Ir. Arif Musthofa, M.T.

ABSTRACT

Development in technology and economy makes electrical load demand increases. The increasing of electrical load must be balanced with the increasing power generated. This makes the scheduling of generating units need to be well determined in order to obtain generation with an optimal price. This final project is developing a software called Powergen that is already used in the calculation of unit commitment and economic dispatch, but taking into account the ramp-rate. Complete enumeration with dynamic programming on the unit commitment is the method which is used in this software. While in the economic dispatch using lambda iteration method. These methods are applied at Delphi is expected to produce unit commitment and economic dispatch calculation software considering the ramp-rate.

Keywords: Delphi, economic dispatch, ramp-rate, unit commitment.

Halaman ini sengaja dikosongkan

Pengembangan Software Unit Commitment And Economic Dispatch dengan Mempertimbangkan Ramp-rate

Nama : Wildan Nurwanto
NRP : 2211100167
Dosen Pembimbing I : Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
Dosen Pembimbing II : Ir. Arif Musthofa, M.T.

ABSTRAK

Kemajuan teknologi dan ekonomi membuat permintaan beban listrik meningkat. Meningkatnya beban listrik harus diimbangi dengan penambahan daya yang dibangkitkan. Hal ini menjadikan penjadwalan unit pembangkit perlu ditentukan dengan baik agar didapatkan pembangkitan dengan harga yang optimal. Tugas Akhir ini mengembangkan *software* Powergen yang sudah biasa digunakan dalam perhitungan *unit commitment* dan *economic dispatch* namun dengan mempertimbangkan *ramp-rate*. Metode yang digunakan adalah *complete enumeration* dengan *forward dynamic programming approach* pada *unit commitment*. Sedangkan pada *economic dispatch* menggunakan metode iterasi lambda. Metode-metode tersebut diterapkan pada Delphi sehingga diharapkan akan menghasilkan *software* perhitungan *unit commitment* dan *economic dispatch* dengan mempertimbangkan *ramp-rate*.

Kata kunci: Delphi, *economic dispatch*, *ramp-rate*, *unit commitment*.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah Robbil 'Alamin, terucap syukur kehadiran Allah SWT atas limpahan rahmat, berkah dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “*PENGEMBANGAN SOFTWARE UNIT COMMITMENT AND ECONOMIC DISPATCH DENGAN MEMPERTIMBANGKAN RAMP-RATE*”. Tujuan dari penyusunan tugas akhir ini adalah sebagai salah satu persyaratan untuk mendapatkan gelar sarjana teknik pada bidang studi Teknik Sistem Tenaga, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah berjasa dalam proses penyusunan tugas akhir ini, yaitu :

1. Allah SWT atas karunia, berkah dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Kedua orang tua penulis, Wawan Gunawan S.T., M.M. dan Dra. Ani Umiyati serta kakak penulis Tia dan adik penulis Zaidan atas dukungan, dorongan semangat, dan doa untuk keberhasilan penulis.
3. Bapak Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D. dan Bapak Ir. Arif Musthofa, MT. selaku dosen pembimbing yang dengan sabar telah memberikan saran, masukan, dan bimbingannya.
4. Anggota kelompok Tugas Akhir pemrograman Delphi yaitu Azwar, Hardi, Adam, Idrus, Rhezal, Aga, dan Rizky yang telah meluangkan waktunya untuk berdiskusi dalam menyelesaikan Tugas Akhir ini.
5. Keluarga besar E-51 ELVN atas kebersamaan, kekeluargaan, dan doanya selama ini.
6. Seluruh keluarga besar Teknik Elektro ITS atas kerjasamanya selama masa kuliah dan proses pengerjaan Tugas Akhir.
7. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu atas inspirasinya dalam pengerjaan Tugas Akhir ini.

Penulis berharap agar buku ini dapat memberikan manfaat bagi banyak pihak, sehingga penulis sangat mengharapkan kritik dan saran membangun dari seluruh pembaca.

Surabaya, Juli 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Halaman

HALAMAN JUDUL	
PERNYATAAN KEASLIAN TUGAS AKHIR	
LEMBAR PENGESAHAN	
ABSTRAK.....	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan	3
1.7 Relevansi.....	4
BAB 2 <i>UNIT COMMITMENT AND ECONOMIC DISPATCH</i> .	5
2.1 Karakteristik Unit Pembangkit.....	5
2.1.1 Karakteristik Pembangkit Termal	5
2.2 <i>Economic dispatch</i>	9
2.2.1 Persamaan Matematis <i>Economic dispatch</i>	10
2.3 <i>Ramp-rate</i>	11
2.4 <i>Unit commitment</i>	12
2.4.1 <i>Minimum Up-Time</i>	12
2.4.2 <i>Minimum Down-Time</i>	13
2.4.3 <i>Start-Up Cost</i>	13
2.5 Iterasi Lambda	13
2.6 <i>Dynamic Programming</i>	15
2.7 Delphi	17

BAB 3 IMPLEMENTASI <i>DYNAMIC PROGRAMMING</i> PADA UNIT COMMITMENT	21
3.1 Algoritma	21
3.2 <i>Dynamic Programming</i> pada <i>Unit Commitment</i>	23
3.3 Argumen <i>Input Unit commitment</i>	24
3.4 Sintaksis Unit Commitment	25
3.5 Iterasi Lambda pada <i>Economic Dispatch</i>	26
3.6 Penggunaan <i>Software</i> Perhitungan <i>Unit Commitment and Economic Dispatch</i>	27
 BAB 4 SIMULASI DAN ANALISIS	 35
4.1 Studi Kasus 1	35
4.2 Studi Kasus 2	37
4.3 Studi Kasus 3	41
 BAB 5 PENUTUP	 49
5.1 Kesimpulan	49
5.2 Saran	49
 DAFTAR PUSTAKA	 51
RIWAYAT HIDUP	53

DAFTAR TABEL

	Halaman
Tabel 3.1 Argumen <i>input</i>	24
Tabel 3.2 Sintaksis.....	25
Tabel 4.1 Data <i>input</i> studi kasus 1	35
Tabel 4.2 Data beban studi kasus 1	36
Tabel 4.3 Hasil simulasi studi kasus 1	36
Tabel 4.4 Data <i>input</i> studi kasus 2	37
Tabel 4.5 Data beban studi kasus 2.....	38
Tabel 4.6 Hasil simulasi studi kasus 2	39
Tabel 4.7 Data <i>input</i> studi kasus 3	42
Tabel 4.8 Data beban studi kasus 3.....	42
Tabel 4.9 Hasil simulasi studi kasus 3	43
Tabel 4.10 Interval yang masih melanggar <i>ramp-rate</i>	44
Tabel 4.11 Hasil simulasi studi kasus 3 dengan <i>ramp-rate</i>	44
Tabel 4.12 Interval yang sudah tidak melanggar <i>ramp-rate</i>	45

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kurva <i>input-output</i> pembangkit termal	6
Gambar 2.2 Kurva karakteristik <i>input-output</i> pembangkit termal uap.....	8
Gambar 2.3 Representasi kurva kenaikan input.....	9
Gambar 2.4 Penyelesaian iterasi lambda	13
Gambar 2.5 Proyeksi lambda.....	14
Gambar 2.6 Contoh dari <i>dynamic programming</i>	16
Gambar 2.7 Tampilan pengerjaan Delphi	18
Gambar 2.8 Tampilan Form Designer	18
Gambar 2.9 Tampilan Object Inspector	19
Gambar 2.10 Tampilan Object TreeView.....	19
Gambar 2.11 Tampilan Code Editor	20
Gambar 3.1 <i>Flowchart</i> penyelesaian Tugas Akhir.....	22
Gambar 3.2 <i>Flowchart unit commitment</i> dengan metode <i>dynamic programming</i>	23
Gambar 3.3 <i>Flowchart</i> iterasi lambda.....	26
Gambar 3.4 Menu utama <i>unit commitment and economic dispatch</i>	27
Gambar 3.5 <i>Editor input</i>	28
Gambar 3.6 <i>Input</i> untuk beban setiap interval	28
Gambar 3.7 <i>Editor</i> untuk proses	29
Gambar 3.8 Tampilan untuk hasil perhitungan.....	30
Gambar 3.9 <i>Main menu</i> EDC.....	31
Gambar 3.10 <i>Input Editor</i> untuk EDC	31
Gambar 3.11 Tampilan untuk memilih <i>ramp-rate</i> dan input beban.....	32
Gambar 3.12 Hasil EDC	32
Gambar 4.1 Grafik pembebanan studi kasus 2.....	41
Gambar 4.2 Total biaya perhitungan dengan <i>ramp-rate</i> dan tidak dengan <i>ramp-rate</i>	46
Gambar 4.3 Pembangkitan unit 6.....	46

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada sistem tenaga listrik, pengaturan penjadwalan unit pembangkit dibutuhkan untuk mendapatkan biaya pembangkitan yang paling ekonomis. Permintaan beban yang terus berubah setiap jam, hari, minggu, bulan, bahkan tahun mengakibatkan perubahan besar daya yang disalurkan pada sisi pembangkitan. Hal ini menyebabkan terjadinya mekanisme penyambungan dan pelepasan unit pembangkit pada sistem kelistrikan yang disebut *unit commitment*. Untuk mendapatkan biaya pembangkitan minimal maka digunakan juga *economic dispatch*.

Dalam sistem penjadwalan unit pembangkit, banyak studi yang berfokus pada aspek ekonomi namun pada studi tersebut berasumsi bahwa perubahan kapasitas pembangkitan mengikuti karakteristik *step function*. Meskipun perhitungan seperti itu menyederhanakan permasalahan, namun hal ini tidak sesuai dengan operasi yang sebenarnya terjadi. Maka diperlukanlah batasan *ramp-rate* untuk menentukan perubahan yang terjadi pada proses pembangkitan [1].

Pada saat ini aplikasi perhitungan *unit commitment* dan *economic dispatch* masih sangat terbatas. Dosen maupun mahasiswa Teknik Elektro ITS biasanya memakai aplikasi PowerGen untuk perhitungan tersebut. Namun PowerGen belum memiliki parameter *ramp-rate* untuk perhitungan *unit commitment and economic dispatch*. Selain itu, program *unit commitment* yang sudah ada hanya dapat memasukkan *input* yang memiliki fungsi linier. Oleh karena itu pada Tugas Akhir kali ini akan mengembangkan *software* tersebut dengan menambahkan *input* nonlinier pada *unit commitment* dan menghitung pembangkitan dalam *unit commitment* tersebut diselesaikan dengan *economic dispatch*. Setelah itu diperiksa kembali apakah hasil yang sudah didapat melebihi batasan *ramp-rate* atau tidak.

1.2 Perumusan Masalah

Permasalahan yang akan dibahas pada Tugas Akhir ini adalah:

1. Bagaimana membuat *software unit commitment* dengan *input* nonlinier orde dua berbasis Delphi?

2. Bagaimana melakukan *economic dispatch* dengan metode iterasi lambda pada *unit commitment*?
3. Bagaimana melakukan pengecekan hasil *unit commitment* agar tetap memenuhi batasan *ramp-rate*?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini sebagai berikut :

1. Pembuatan *software* menggunakan Delphi.
2. *Losses* diabaikan.
3. Metode yang digunakan untuk menghitung *economic dispatch* dengan *input* nonlinier adalah iterasi lambda.
4. Metode yang digunakan untuk menghitung *unit commitment* adalah *dynamic programming*.
5. Data beban tidak melebihi kemampuan pembangkitan total pembangkit yang aktif.
6. Sistem dalam keadaan stabil.
7. Data *input* hanya menggunakan satu jenis *start up cost*.
8. Tidak mempertimbangkan biaya padam.

1.4 Tujuan

Penulisan Tugas Akhir ini bertujuan untuk:

1. Membuat *software unit commitment* dengan *input* nonlinier orde dua.
2. Membuat *software unit commitment and economic dispatch* dengan memperhatikan *ramp-rate*.
3. Melihat pengaruh *ramp-rate* pada perhitungan *unit commitment and economic dispatch*.

1.5 Metodologi

Pada Tugas Akhir ini menggunakan metodologi sebagai berikut:

1. Studi Literatur
Studi literatur dilakukan dengan membaca *text book*, jurnal, dan juga Tugas Akhir sebelumnya. Dalam studi literatur dipelajari teori pendukung seperti *unit commitment*, *economic dispatch*, *dynamic programming*, metode iterasi lambda, dan pemrograman Delphi.
2. Pengumpulan Data

Data yang diperlukan yaitu data parameter pembangkit dan data beban. Data pembangkit antara lain data fungsi biaya, kemampuan minimal dan maksimal pembangkitan, dan *ramp rate* masing-masing pembangkit. Sedangkan data beban berisi data pembebanan dengan interval tertentu.

3. Pengembangan *Software* berbasis Delphi
Software yang dikembangkan adalah PowerGen. *Software* ini diberi tambahan *input* sehingga dapat menghitung data pembangkit yang masukannya nonlinier.
4. Simulasi dan Analisis
Simulasi dilakukan dengan tiga kondisi yang berbeda, kemudian melakukan analisis terhadap hasil simulasi.
5. Penulisan Buku
Penulisan buku dilakukan sebagai hasil dari kesimpulan dari Tugas Akhir ini. Kesimpulan tersebut adalah jawaban dari permasalahan yang telah dianalisis.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun dengan sistematika sebagai berikut:

- Bab I PENDAHULUAN
Bab ini membahas tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, dan sistematika penulisan Tugas Akhir ini.
- Bab II *UNIT COMMITMENT* DAN *ECONOMIC DISPATCH* DENGAN BATASAN *RAMP-RATE*
Bab ini berisi tentang teori *unit commitment*, *economic dispatch*, *dynamic programming*, iterasi lambda, dan Delphi sebagai basis programnya.
- Bab III *IMPLEMENTASI DYNAMIC PROGRAMMING PADA UNIT COMMITMENT*
Bab ini berisi mengenai penerapan metode iterasi lambda pada *economic dispatch* dan *dynamic programming* untuk *unit commitment*.
- Bab IV *SIMULASI DAN ANALISIS*
Bab ini berisi hasil simulasi program *unit commitment* dan *economic dispatch* berbasis Delphi serta analisis hasil *unit commitment* dari tiga kasus yang berbeda.
- Bab V PENUTUP
Bab ini berisi kesimpulan dan saran.

1.7 Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat memberikan kontribusi sebagai berikut:

1. Dapat memberikan manfaat dalam kemudahan melakukan perhitungan *unit commitment* dan *economic dispatch* dengan memperhatikan batasan-batasan seperti *minimum up-time*, *minimum down-time*, dan juga *ramp-rate*.
2. Software *unit commitment* and *economic dispatch* ini dapat digunakan untuk mata kuliah Operasi Optimum Sistem Tenaga Listrik.
3. Dapat menambah keilmuan dan teknologi dalam bidang operasi optimalisasi pembangkitan sistem tenaga listrik.
4. Dapat menjadi referensi Tugas Akhir mahasiswa yang akan mengambil Tugas Akhir untuk mengembangkan aplikasi perhitungan *unit commitment* and *economic dispatch* dengan permasalahan-permasalahan yang berbeda.

BAB 2

UNIT COMMITMENT AND ECONOMIC DISPATCH

2.1 Karakteristik Unit Pembangkit

Setiap unit pembangkit memiliki karakteristiknya masing-masing. Perbedaan karakteristik tersebut menyebabkan peran pembangkit beban menjadi berbeda. Secara umum peran unit pembangkit dibedakan menjadi pembangkit pemikul beban dasar (*base load*), pemikul beban menengah (*load follower*), dan pemikul beban puncak (*peak load*).

Pembangkit pemikul beban dasar (*base load*) memiliki karakteristik kurang fleksibel. Pembangkit ini tidak dapat dihidupkan atau dimatikan dalam waktu singkat serta lambat dalam menaikkan atau menurunkan pembebanan. Pembangkit *base load* yang berkemampuan pembangkitan yang besar memiliki biaya produksi yang lebih murah jika dibandingkan dengan kelompok pembangkit lain. Pembangkit *base load* beroperasi maksimum sepanjang pembangkit tersebut siap serta memang dibutuhkan oleh sistem.

Pembangkit yang lebih fleksibel namun lebih mahal sedikit daripada pembangkit *base load* adalah pembangkit *load follower*. Contoh dari pembangkit *load follower* adalah PLTGU gas dan PLTU minyak.

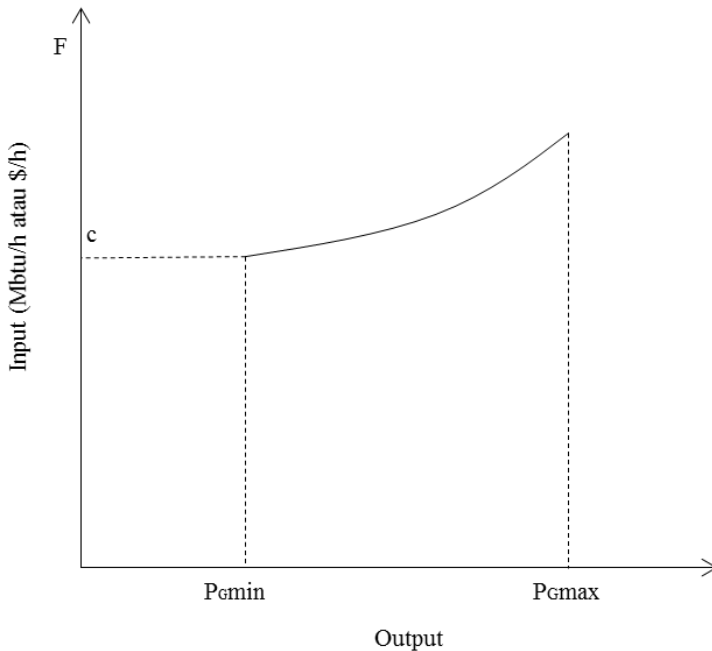
Sedangkan untuk pembangkit yang difungsikan sebagai pembangkit *peak load* harus dapat fleksibel dalam kecepatan perubahan pembebanan maupun operasi hidup dan mati pembangkit. Contoh pembangkit *peak load* adalah PLTG minyak, PLTD serta PLTA waduk. Operasi dari pembangkit *peak load* kurang dari 100 MW.

2.1.1 Karakteristik Pembangkit Termal

Karakteristik *input-output* konsumsi bahan bakar pembangkit adalah dasar penyusunan fungsi biaya pada pembangkit termal. Secara umum, karakteristik *input-output* pembangkit termal berbentuk Btu/jam *input* ke unit generator. Biaya pembangkitan didapat dari perkalian biaya kalori yang terkandung dalam bahan bakar dengan kebutuhan kalori tiap jam dari generator. Hasil daya yang dapat dibangkitkan ditampilkan dengan P_G . Pada Gambar 2.1 menunjukkan

bahwa kurva karakteristik *input-output* dari pembangkit termal memiliki batas minimal pembangkitan dan maksimal pembangkitan dari daya *output* yang diproduksi.

Biaya operasi dari pembangkit termal selain biaya bahan bakar yang dikonsumsi juga meliputi biaya tenaga kerja, biaya pemeliharaan, biaya transportasi, dan lain-lain. Biaya-biaya tersebut tidak ditampilkan secara langsung sebagai fungsi biaya dari daya *output* yang dihasilkan generator. Berdasarkan permasalahan tersebut, biaya-biaya tersebut diasumsikan sebagai bagian dari *fixed cost* dari biaya operasi, dan diabaikan dalam proses perhitungan optimisasi pada Tugas Akhir ini.



Gambar 2.1 Kurva *input-output* pembangkit termal

Pembangkit termal sederhana terdiri atas *boiler*, turbin uap, dan generator. *Input* dari *boiler* adalah bahan bakar dan *output* berupa uap. Hubungan dari *input-output boiler* dapat ditampilkan dalam bentuk *convex curve*. Uap adalah *input* dari turbin dan keluarannya

adalah daya listrik. Kita dapat merepresentasikan karakteristik dari keseluruhan sistem dengan menggabungkan karakteristik *input-output* dari *boiler* dan turbin generator.

Kondisi teknis atau faktor lain pada *boiler* dan turbin menyebabkan batas minimum dari daya *output* generator harus diperhatikan. Pada turbin, kondisi teknis yang dimaksud adalah kondisi *shell* serta perbedaan suhu metal motor, *exhaust hood temperature*, *rotor expansion* dan *shell expansion*. Pada *boiler*, kondisi teknis tersebut disebabkan oleh *fuel combustion stability* and *valve*. Sedangkan batas maksimal dari daya *output* suatu pembangkit ditentukan dari desain kapasitas *boiler* dan turbin generator.

Untuk menentukan karakteristik *input-output*, pendekatan atau linierisasi dari biaya bahan bakar yang masuk generator terhadap daya *output* yang diperoleh berdasarkan beberapa cara, yaitu:

1. Berdasarkan percobaan efisiensi dari pembangkit.
2. Berdasarkan data historis tentang operasi dari suatu unit generator.
3. Berdasarkan data desain dari unit generator yang dikeluarkan oleh pabrikan generator.

Biaya pembangkitan energi listrik dari suatu pembangkit itulah yang didapat dari persamaan karakteristik yang diperoleh. Persamaan karakteristik *input-output* pembangkitan termal secara umum ditampilkan dalam persamaan orde dua. Namun persamaan tersebut juga dapat direpresentasikan dalam orde lebih dari dua dan dapat menjadi lebih tidak linier apabila perhitungan yang lain diperhitungkan seperti *valve-point effect*.

Kita membutuhkan data karakteristik *input-output* dari suatu pembangkit termal untuk menganalisis permasalahan mengenai operasi dalam sistem tenaga, terutama masalah operasi ekonomis. *Gross input* dan *net output* dibahas untuk mendefinisikan karakteristik unit tersebut.

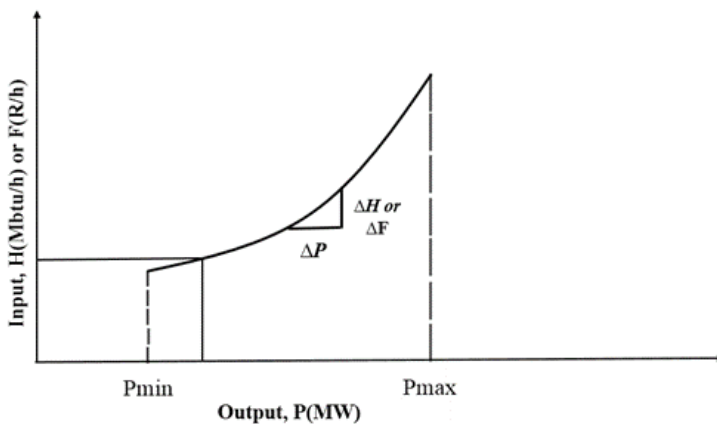
Gross unit pembangkit menunjukkan total *input* dan diukur dalam dolar per jam, atau ton bahan bakar per jam atau kubik gas per jam atau dalam bentuk yang lain. Sedangkan *net output* unit pembangkit yaitu *output* daya listrik yang tersedia untuk penggunaan dalam sistem tenaga,

Untuk mendefinisikan karakteristik unit turbin uap digunakan beberapa konstanta H dan F. H adalah Btu per jam *input* panas

(Mbtu/h). Sedangkan F adalah biaya bahan bakar dikalikan H , yaitu R per jam (R/h) *input* pada unit untuk bahan bakar.

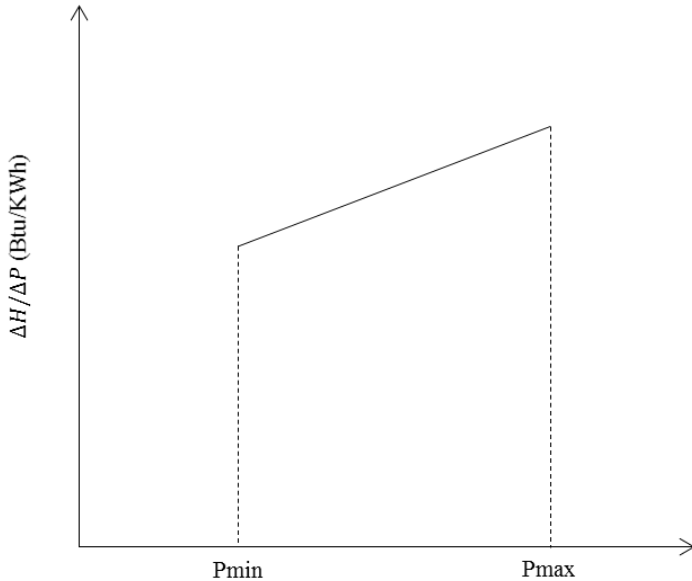
Biaya operasional R per jam suatu unit terdiri atas biaya operasional dan biaya pemeliharaan. Biaya pekerja dimasukkan sebagai bagian dari biaya operasi jika biaya ini dapat digambarkan sebagai fungsi *output* unit. *Output* unit dinotasikan dengan P .

Karakteristik *input-output* dapat dilihat pada Gambar 2.2. *Input* unit dapat diterjemahkan dalam bentuk energi panas maupun biaya total per jam. Sedangkan *output* adalah *output* daya listrik dari unit tersebut.



Gambar 2.2 Kurva karakteristik *input-output* pembangkit termal uap

Pada Gambar 2.3 menunjukkan karakteristik kenaikan panas (*incremental heat*) dari unit pembangkit uap. Karakteristik ini adalah karakteristik kemiringan dari karakteristik *input-output* ($\Delta H/\Delta P$ atau $\Delta F/\Delta P$). Satuan dari *incremental heat* ini adalah Btu per KWh (R per KWh).



Gambar 2.3 Representasi kurva kenaikan *input*

2.2 *Economic dispatch*

Economic dispatch adalah suatu pembagian pembebanan pada unit-unit pembangkit secara optimal ekonomi yang ada dalam sistem, pada harga beban sistem tertentu [2]. Oleh karena beban sistem yang selalu berubah dari waktu ke waktu, maka untuk perhitungan *economic dispatch* tersebut juga dilakukan setiap beban tersebut.

Untuk menyelesaikan permasalahan *economic dispatch*, ada beberapa metode yang dapat digunakan, diantaranya yaitu *Lagrange Relaxation*, iterasi lambda, *base point*, dan lain-lain. Pada Tugas Akhir ini digunakan iterasi lambda untuk menyelesaikan permasalahan *economic dispatch*.

Pada pembangkitan energi listrik, biaya operasi adalah biaya yang paling dominan pada sistem operasi tenaga listrik. Biaya utama lainnya adalah biaya pembangunan fasilitas dan biaya kepemilikan.

Komponen yang berpengaruh dalam biaya operasi adalah biaya bahan bakar (*fuel cost*) dan setiap unit pembangkit memiliki

karakteristik biaya bahan bakar yang berbeda sesuai dengan jenis bahan bakar dan efisiensi tiap pembangkit. Biaya produksi energi listrik dipengaruhi pengoptimalan biaya produksi dengan mempertimbangkan *fuel cost*. Oleh karena itu, meskipun pada kondisi normal kapasitas total dari pembangkit lebih besar daripada daya beban dan *transmission losses*, penjadwalan pembebanan pembangkit menjadi penting untuk menekan biaya produksi.

Economic dispatch memiliki tujuan utama yaitu meminimalkan konsumsi bahan bakar dari setiap unit pembangkit pada keseluruhan sistem dengan menentukan daya *output* setiap unit pembangkit. Penentuan daya yang dikeluarkan pada setiap generator tersebut bergantung pada batasan-batasan tertentu (*constraint*).

2.2.1 Persamaan Matematis *Economic dispatch*

Optimasi yang dilakukan pada permasalahan *economic dispatch* merupakan optimasi dari sisi biaya bahan baku pembangkitan atau *fuel cost*. Biaya bahan baku pembangkitan ini memiliki karakteristik nonlinier. Bentuk dari persamaan biaya pembangkitan adalah persamaan *polynomial* orde dua yang dapat dituliskan sebagai berikut:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (2.1)$$

keterangan:

F_i = biaya pembangkitan pada pembangkitan ke- i (\$)

P_i = Daya *output* unit pembangkit i (MW)

Variabel a , b , c merupakan konstanta biaya operasi dari suatu pembangkit. Koefisien c menunjukkan biaya operasi pembangkit ketika tidak memproduksi energi listrik.

Berdasarkan persamaan (2.1), dapat diketahui bahwa antara daya yang dibangkitkan generator nonlinier terhadap biaya pembangkitan. Kombinasi daya *output* yang dibangkitkan oleh masing-masing generator pada sistem harus memenuhi kebutuhan daya dari sistem energi listrik (*equality constraint*) dan memenuhi batas minimum serta maksimum dari daya yang dapat dibangkitkan oleh generator (*inequality constraint*). Parameter-parameter *equality constraint* dan *inequality constraint* dapat direpresentasikan dalam persamaan berikut

$$\text{Min } \Sigma F_i(P_i) = \text{Min } \Sigma a_i P_i^2 + b_i P_i + c_i \quad (2.2)$$

$$P_{Gi} \min \leq P_G \leq P_{Gi} \max \quad (2.3)$$

P_{Gi} adalah besarnya daya yang dibangkitkan generator ke- i atau persamaan (2.3) ini disebut dengan *inequality constraint*.

$$\Sigma P_i = P_d + P_L \quad (2.4)$$

Keterangan:

P_d = daya permintaan konsumen (MW)

P_L = rugi daya yang terjadi pada jaringan transmisi (MW)

Persamaan (2.4) disebut dengan *equality constraint*. *Transmission losses* atau P_L pada persamaan (2.4) dipengaruhi oleh besarnya aliran daya yang mengalir pada jaringan transmisi. Berdasarkan persamaan tersebut juga dapat disimpulkan bahwa besarnya kombinasi daya *output* yang dibangkitkan generator mempengaruhi besar aliran daya yang mengalir pada jaringan transmisi dan menentukan besar *transmission losses* yang terjadi.

2.3 Ramp-rate

Ramp-rate adalah kemampuan generator untuk melakukan peningkatan (*up-rate*) atau penurunan (*down-rate*) pembangkitan. Setiap unit pembangkit memiliki karakteristik *ramp-rate* yang berbeda. Oleh karena itu perhitungan *ramp-rate* membutuhkan fungsi tertentu untuk hasil yang optimal.

Fungsi *ramp-rate* dapat direpresentasikan dalam persamaan sebagai berikut:

$$P_i(t-1) - P_i(t) \leq DR_i \quad (2.5)$$

$$P_i(t-1) - P_i(t) \leq UR_i \quad (2.6)$$

Jika persamaan (2.5) dan persamaan (2.6) disatukan, maka akan mendapatkan batasan baru, yaitu:

$$P_i(t - 1) - DR_i \leq P_i(t) \leq DR_i + P_i(t - 1) \quad (2.7)$$

Keterangan:

$P_i(t)$ = pembangkitan unit i pada interval ke-t (MW)
 $P_i(t - 1)$ = pembangkitan unit i pada interval ke t-1 (MW)

2.4 Unit commitment

Kebutuhan pemakaian energi (beban) listrik mengikuti siklus kegiatan manusia sehari-hari. Naik atau turunnya pemakaian energi listrik selalu diimbangi oleh pembangkitan energi listrik dalam sistem. Untuk mengikuti siklus pembangkitan energi listrik tersebut dilakukan penjadwalan unit yang *commit (on)* dan unit yang *off* dalam siklus waktu tertentu.

Dalam penjadwalan pembangkit tersebut, dipertimbangkan kondisi optimal ekonomi, selain itu harus memenuhi batasan-batasan teknis dalam pengoperasian pembangkitan di dalam sistem tenaga. Penjadwalan tersebut dinamakan dengan *unit commitment* [1].

Pada *unit commitment* dikenal dengan kombinasi unit pembangkit. Kombinasi unit pembangkit merupakan kombinasi *on/off* dari beberapa unit yang ada dalam sistem. Jika ada N pembangkit, maka dapat dihitung jumlah kombinasi yang mungkin adalah $2^N - 1$. Dari kombinasi yang ada akan dipilih kombinasi yang dijadwalkan. Evaluasi pemilihan kombinasi yang masuk dalam jadwal pembangkitan dilakukan dengan menghitung biaya optimum (*economic dispatch*) untuk setiap kombinasi sehingga dapat dibandingkan biaya optimum terendah dari kombinasi-kombinasi *on/off* unit pembangkit pada beban tertentu.

Pada *unit commitment* ada beberapa *constraint* yang dipertimbangkan, yaitu:

2.4.1 Minimum Up-Time

Minimum up-time adalah interval waktu minimum suatu unit yang sedang *on* (terhubung dengan sistem) tidak boleh dilepas (*off*) kembali sebelum melewati batas *up-time*.

2.4.2 Minimum Down-Time

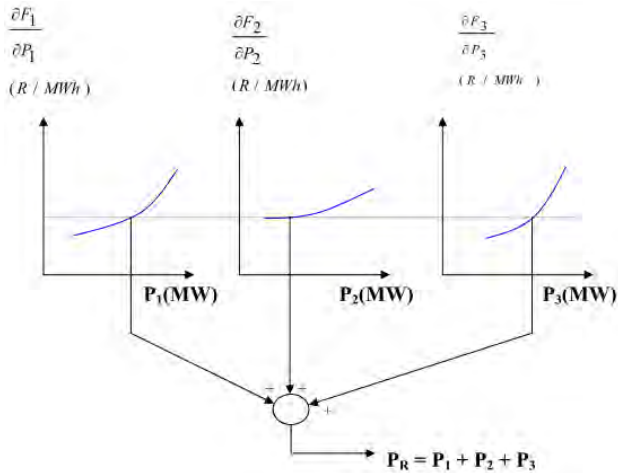
Minimum down-time adalah interval waktu minimum suatu unit dalam keadaan *decommit (off)* tidak boleh dihubungkan ke sistem (*on*) sebelum melewati batas *down-time*.

2.4.3 Start-Up Cost

Start-up cost adalah biaya yang diperlukan oleh pembangkit untuk start dari keadaan tidak beroperasi (terhubung ke sistem tenaga listrik). *Start-up cost* ada dua macam, yaitu *cold start* dan *hot start* (biaya start panas).

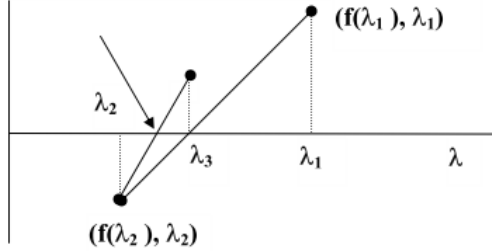
2.5 Iterasi Lambda

Metode iterasi lambda digunakan untuk menyelesaikan persamaa *economic dispatch*. Iterasi lambda dapat direpresentasikan seperti Gambar 2.4 dan Gambar 2.5.



Gambar 2.4 Penyelesaian iterasi lambda

dicapai : $|\varepsilon| < \text{Toleransi}$



Gambar 2.5 Proyeksi lambda

Lambda berasal dari persamaan lagrangian:

$$\frac{\partial F_i}{\partial P_i} = \lambda \left(1 - \frac{\partial P_{loss}}{\partial P_i} \right) \quad (2.8)$$

$$\lambda = \frac{\partial F_i}{\partial P_i} \left(\frac{1}{1 - \frac{\partial P_{loss}}{\partial P_i}} \right) \quad (2.9)$$

$$\lambda = \frac{\partial F_i}{\partial P_i} P f_i \quad (2.10)$$

Lambda start direpresentasikan pada persamaan (2.13):

$$\lambda_{min} = \min \left(\frac{\partial F_i}{\partial P_i} P f_i, i = 1 \dots \text{Jumlah Generator} \right) \quad (2.11)$$

$$\lambda_{max} = \max \left(\frac{\partial F_i}{\partial P_i} P f_i, i = 1 \dots \text{Jumlah Generator} \right) \quad (2.12)$$

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2} \quad (2.13)$$

Jika nilai pembangkitan belum sesuai, perlu dilakukan iterasi. Untuk melakukan iterasi diperlukan $\Delta\lambda$ yang dapat direpresentasikan pada persamaan (2.14)

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} \quad (2.14)$$

$$P_{target} = P_{loss} + P_{load} \quad (2.15)$$

Setelah itu, jika nilai yang didapatkan pada iterasi sama dengan persamaan (2.16), maka untuk nilai lambda selanjutnya menggunakan persamaan (2.17).

$$\sum_i^n P_i - P_{target} > 0, \quad n = \text{Jumlah Generator} \quad (2.16)$$

$$\lambda = \lambda_{sebelum} - \Delta\lambda \quad (2.17)$$

Namun jika nilai yang didapatkan pada iterasi sama dengan persamaan (2.18), maka untuk nilai lambda selanjutnya menggunakan persamaan (2.19).

$$\sum_i^n P_i - P_{target} < 0, \quad n = \text{Jumlah Generator} \quad (2.18)$$

$$\lambda = \lambda_{sebelum} + \Delta\lambda \quad (2.19)$$

Nilai lambda akan terus diiterasi hingga mendapatkan nilai seperti pada persamaan (2.20).

$$\sum_i^n P_i - P_{target} = 0, \quad n = \text{Jumlah Generator} \quad (2.20)$$

Pada Tugas Akhir ini nilai *error* (ε) ditentukan sebesar 0.01. Ketika diimplementasikan ke dalam persamaan, nilai lambda akan berhenti melakukan iterasi hingga mendapatkan nilai seperti pada persamaan (2.21)

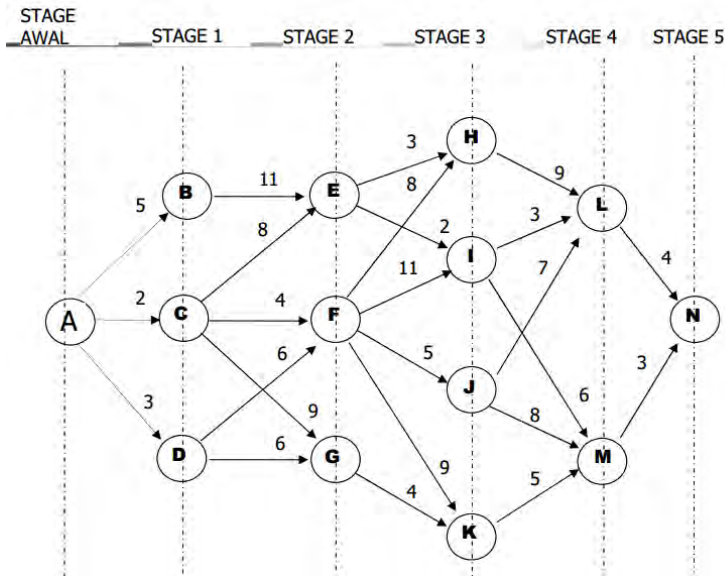
$$\sum_i^n P_i - P_{target} = \varepsilon, \quad n = \text{Jumlah Generator} \quad (2.21)$$

Setiap nilai P_i harus memenuhi syarat *inequality constrain* dan batasan *ramp-rate* yang digunakan untuk penyelesaian masalah ED pada Tugas Akhir ini.

2.6 Dynamic Programming

Dr. Richard Bellman menemukan metode *dynamic programming* untuk memecahkan berbagai persoalan kontrol dan optimasi dinamis. Metode ini sangat berguna untuk memecahkan berbagai persoalan dan mengurangi perhitungan dalam menemukan trajektori optimal.

Untuk menyelesaikan *unit commitment* digunakan *dynamic programming* dengan penelusuran ke depan (forward approach). Gambar 2.6 adalah contoh dari *dynamic programming*.



Gambar 2.6 Contoh dari *dynamic programming*

Keterangan:

State : terminal-terminal dengan kondisi tertentu.

Stage : kumpulan dari *state* pada level tertentu.

Gambar 2.6 menunjukkan model pembiayaan suatu proses yang diawali dari *state* A sampai ke *state* N dengan melalui banyaknya pilihan jalur pembiayaan. Terdapat lima *stage*, setiap *stage* memiliki beberapa *state*.

Pada *dynamic programming*, persamaan dapat direpresentasikan sebagai berikut:

$$F_n(S, x_n) = C_{S, x_n} + F_{n-1}^*(x_{n-1}) \quad (2.22)$$

Keterangan:

x_n : *stage* ke- n

S : variabel *state* yang berada di *stage* x_{n-1}

C_{S, x_n} : biaya dari *state* S ke *state* yang berada di x_n

$F_n(S, x_n)$: biaya kumulatif untuk sampai ke <i>state</i> yang berada di x_n melewati variable <i>state</i> S
$F_n^*(x_n)$: biaya kumulatif termurah untuk sampai ke <i>state</i> yang berada di x_n melewati variable <i>state</i> S
$F_{n-1}^*(x_{n-1})$: biaya kumulatif untuk sampai ke <i>state</i> yang berada di x_{n-1}

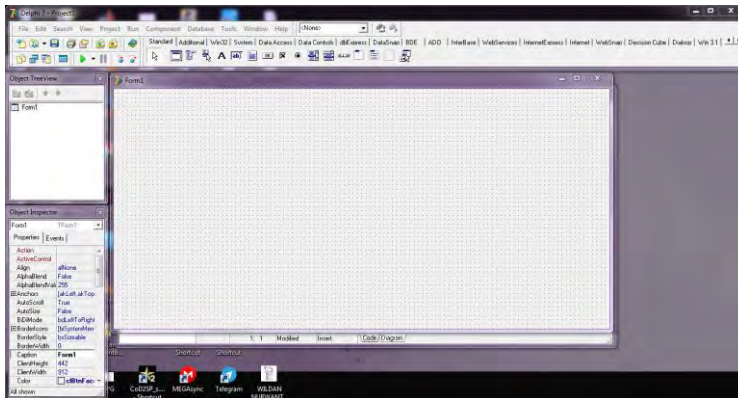
2.7 Delphi

Delphi adalah *Integrated Development Environment* (IDE) untuk mengembangkan aplikasi *desktop*, *web*, maupun *mobile*. Produk ini awalnya dikembangkan oleh CodeGear sebagai divisi pengembangan perangkat lunak milik Embarcadero, divisi tersebut sebelumnya adalah milik Borland. Bahasa Delphi, atau dikenal pula sebagai object pascal pada mulanya ditujukan hanya untuk Microsoft Windows, namun saat ini telah mampu digunakan untuk mengembangkan aplikasi untuk Mac OS X, iOS, Microsoft .NET framework.

Pada Tugas Akhir ini menggunakan Delphi untuk mengaplikasikan rumus-rumus yang berkaitan dengan permasalahan. Delphi dapat menghasilkan aplikasi perhitungan yang memiliki *interface* yang mudah dipahami. Delphi memiliki beberapa kelebihan diantaranya:

1. Kemudahan penyusunan *User Interface*, Delphi berkomitmen untuk menjadi *Rapid Application Development* (RAD). RAD adalah bagaimana mempercepat perkembangan aplikasi.
2. Bahasa *Object Pascal* merupakan salah satu jenis dari bahasa pascal dengan sejumlah penambahan, terutama terkait dengan dengan konsep *Object Oriented Programing* (OOP). Dengan salah satu kelebihan bahasa Pascal yang mudah dipahami dan tidak terlalu kompleks.
3. *Native Code*, hasil compile Delphi adalah kode *native* untuk window 32. Ini berarti *file exe* yang dihasilkan oleh kompiler akan langsung dijalankan oleh mesin tanpa melalui *software* lain seperti *Virtual Machine* (VM). Hasil dari Delphi adalah *file exe* tunggal, tanpa perlu file-file lainnya.[4]

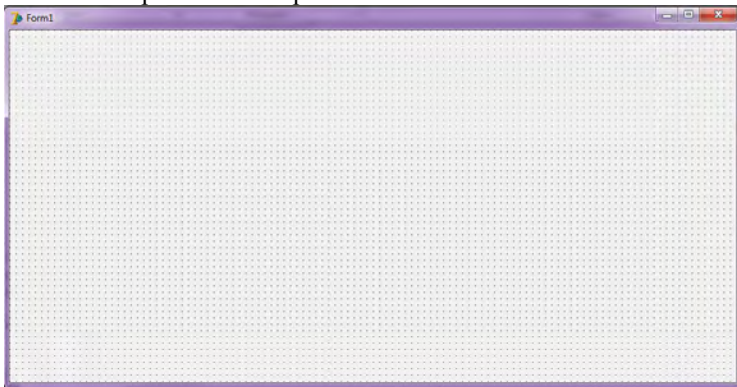
IDE pada Delphi memiliki semua alat yang diperlukan untuk perancangan *software*. Gambar 2.7 merepresentasikan tampilan Delphi ketika buka.



Gambar 2.7 Tampilan pengerjaan Delphi

Bagian penting pada pemrograman Delphi diantaranya:

1. *Form Designer* atau *Form*, merupakan jendela kosong yang digunakan untuk merancang suatu *User Interface* (UI) dalam perancangan aplikasi yang sedang dibuat. *Form* direpresentasikan pada Gambar 2.8.



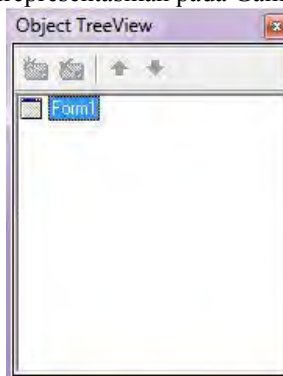
Gambar 2.8 Tampilan *Form Designer*

2. *Object Inspector* digunakan untuk mengatur dan memeriksa sekumpulan *property* yang berada di dalam *Form* untuk mendapatkan tampilan sesuai yang diinginkan. *Object Inspector* direpresentasikan pada Gambar 2.9.



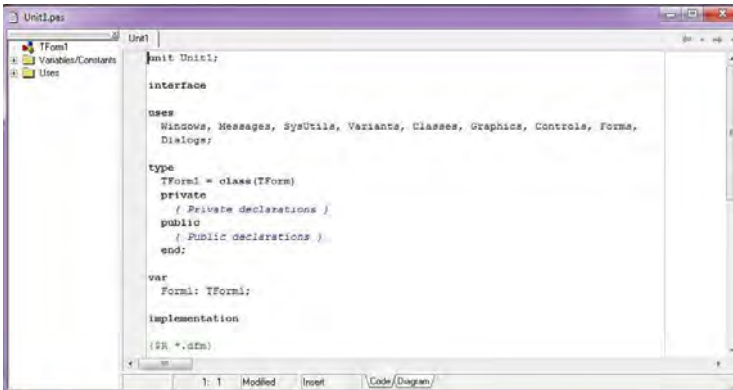
Gambar 2.9 Tampilan Object Inspector

3. *Object TreeView* digunakan untuk menampilkan dan mengubah hubungan antar komponen dalam *Form*. *Object TreeView* direpresentasikan pada Gambar 2.10.



Gambar 2.10 Tampilan *Object TreeView*

4. *Code Editor* adalah tempat penulisan dan memasukkan logika pemrograman. *Code Editor* direpresentasikan pada Gambar 2.11.



Gambar 2.11 Tampilan *Code Editor*

BAB 3

IMPLEMENTASI *DYNAMIC PROGRAMMING* PADA *UNIT COMMITMENT*

Pada Tugas Akhir kali ini *dynamic programming* digunakan untuk menyelesaikan permasalahan dalam perhitungan *unit commitment*, sedangkan untuk pembebanan optimalnya digunakan iterasi lambda. Pengaruh dari batasan *ramp-rate* dikoreksi sesaat setelah pengecekan hasil akhir *unit commitment* dengan pembebanannya. Jika *ramp-rate* sudah terpenuhi, perhitungan sudah selesai. Jika *ramp-rate* belum terpenuhi, maka dilakukan perhitungan ulang dengan *economic dispatch* pada interval waktu tersebut.

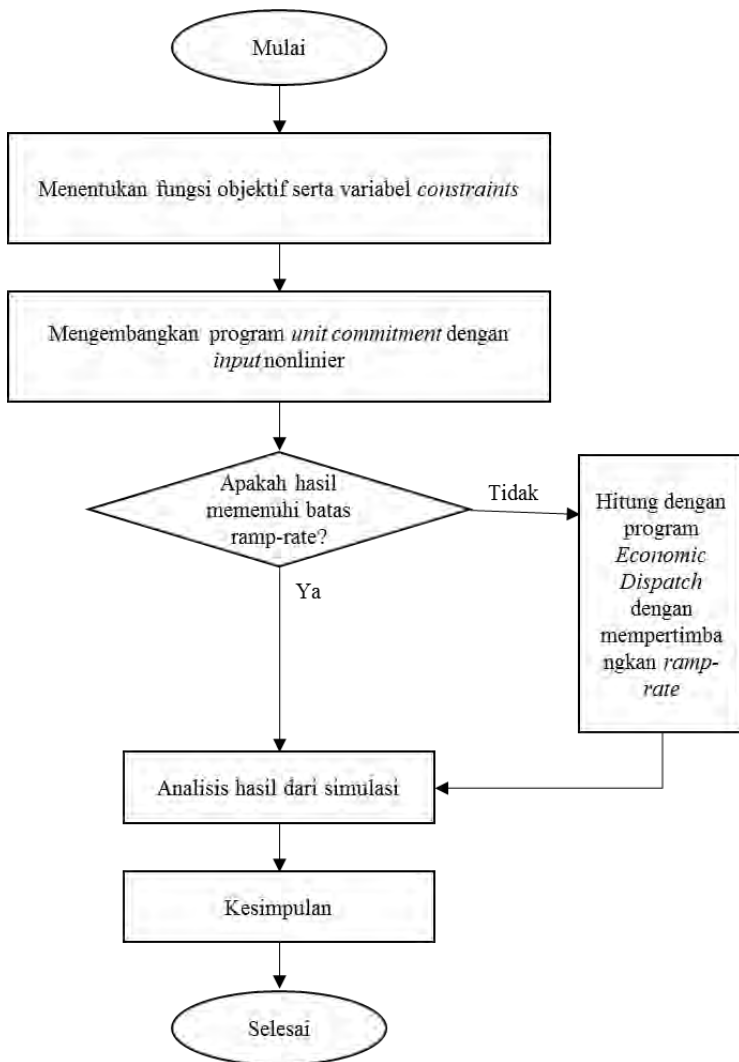
Pengolahan data dan simulasi menggunakan aplikasi perhitungan yang dibuat dengan menggunakan program Delphi. Alur penyelesaian dan penerapan metode dijelaskan pada subbab 3.1.

3.1 Algoritma

Alur Tugas Akhir dimulai dengan menentukan fungsi objektif dan juga variabel *constraint*. Setelah itu program *unit commitment* dikembangkan sehingga dapat digunakan *input* nonlinier. Program *unit commitment* sebelumnya hanya dapat memasukkan data *input* unit pembangkit linier saja, sehingga saat ini sudah dapat digunakan data *input* unit pembangkit nonlinier.

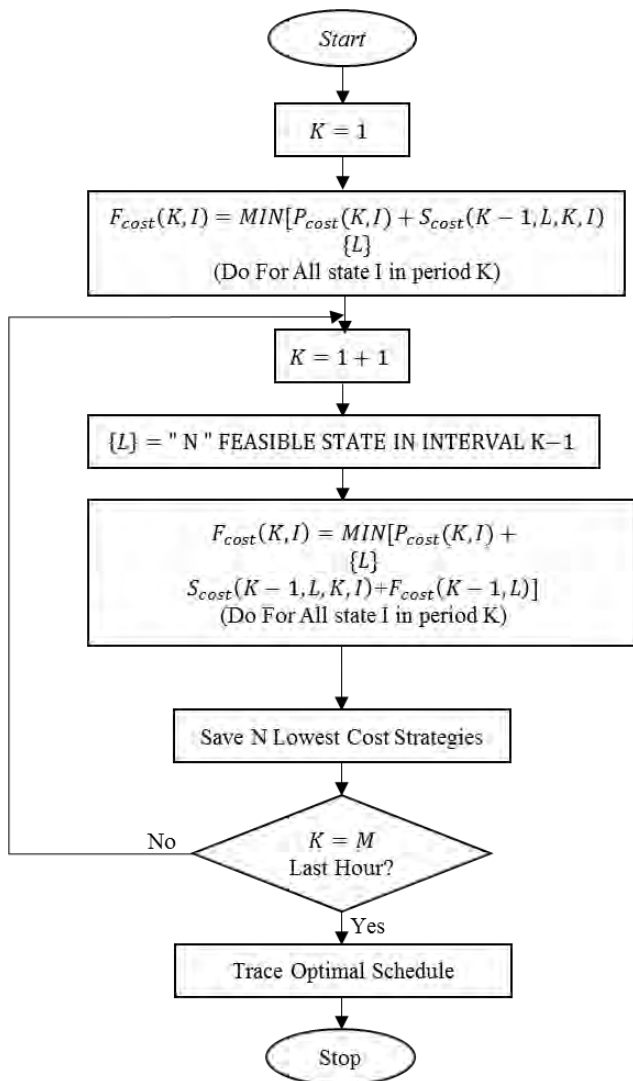
Hasil dari program *unit commitment* ini adalah penjadwalan unit yang *commit (on)* dan juga unit yang *off*. Selain itu, hasil dari program *unit commitment* juga dapat menampilkan pembebanan unit *on* tersebut. Langkah selanjutnya adalah memastikan apakah pembebanan tersebut sudah sesuai dengan *ramp-rate* atau belum. Jika sudah maka hasilnya dapat dianalisis. Jika belum sesuai *ramp-rate*, maka langkah selanjutnya adalah menghitung kembali hasil simulasi dengan program *economic dispatch*. Program *economic dispatch* tersebut sudah dapat menghitung efek dari batasan *ramp-rate*. Perlu dilakukan pengecekan setiap interval waktu untuk memastikan apakah hasil dari *unit commitment* sudah memenuhi batasan *ramp-rate* atau belum. Kemudian hasil akhir dari simulasi program dapat dianalisis dan disimpulkan sebagai kesimpulan dari Tugas Akhir ini.

Flowchart dari algoritma penyelesaian Tugas Akhir ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Flowchart penyelesaian Tugas Akhir

3.2 Dynamic Programming pada Unit commitment



Gambar 3.2 Flowchart unit commitment dengan metode dynamic programming

Metode *dynamic programming* yang digunakan pada *unit commitment* dapat dilihat pada Gambar 3.2.

Keterangan:

- State* : berisikan kombinasi unit-unit pembangkit yang beroperasi dan tidak beroperasi.
- Stage* : level beban pada jam tertentu (interval waktu). Di dalam *stage* terdapat beberapa *state*.
- N Feasible state* : adalah N buah *state* pada *stage* sebelumnya yang dipilih dengan biaya akumulatif termurah dimulai dari urutan yang termurah.
- K* : menyatakan *stage*/jam ke sekian untuk level-level beban tertentu
- $F_{cost}(K, I)$: biaya komulatif yang diperlukan untuk sampai ke *state* I pada level K
- $F_{cost}(K, I)$: biaya produksi pada *state* I untuk level *stage* K
- $S_{cost}(K - 1, L, K, I)$: biaya transisi dari *state* L pada level K-1 ke *state* I pada level K
- X* : jumlah *state* yang harus diperiksa pada setiap interval waktu (pada setiap *stage*)

3.3 Argumen *Input Unit commitment*

Tabel 3.1 Argumen *input*

Argumen	Keterangan
UNTMAX[i]	Sebagai <i>input</i> dari batas <i>maximum</i> pembangkitan unit (Pmax)
UNTMIN[i]	Sebagai <i>input</i> dari batasan <i>minimum</i> pembangkititan unit (Pmin)
UNTUPMIN[i]	Sebagai <i>input</i> dari waktu minimum up time
UNTDOWNMIN[i]	Sebagai <i>input</i> dari waktu minimum down time
inpa[i]	Sebagai <i>input</i> dari koefisien a (orde 2) unit pembangkit
inpb[i]	Sebagai <i>input</i> dari koefisien b (orde 1) unit pembangkit
inpc[i]	Sebagai <i>input</i> dari koefisien c (orde 0) unit pembangkit
UNTSTARTCOST [i]	Sebagai <i>input</i> dari <i>start up cost</i>

Tabel 3.1 Argumen *Input* (lanjutan)

Argumen	Keterangan
UNTSTATINI[i]	Sebagai <i>input</i> dari <i>initial status</i> unit pembangkit
Fuelcost[i]	Sebagai <i>input</i> nilai dari <i>fuelcost</i> yang digunakan untuk persamaan $Fi(Pi(t)) = Hi(Pi(t)) \times \text{fuelcost } i$
SYSLOAD [i]	Sebagai <i>inputan</i> nilai beban (<i>Pload</i>) di setiap periode

3.4 Sintaksis Unit Commitment

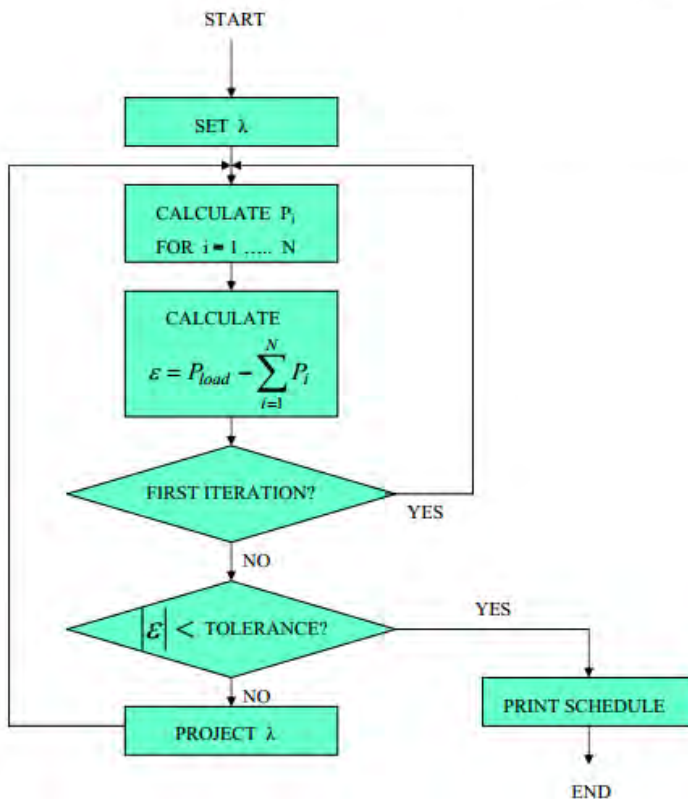
Sintaksis program adalah perintah yang digunakan untuk melakukan pemanggilan program dengan argument *input* yang telah ditentukan.

Tabel 3.2 Sintaksis

Sintaksis	Keterangan
datadump	Digunakan sebagai memasukkan semua data awal perhitungan di setiap periode
EDC	Sebagai prosedur untuk mendapatkan pembebanan optimal untuk tiap kombinasi pada interval waktu tertentu. <i>Economic dispatch</i> ini menggunakan metode iterasi lambda.
invers_ihr_ftn	Mencari nilai pembangkitan setiap unit ketika didapatkan nilai lambda $\frac{\partial Fi}{\partial Pi} = \lambda \left(1 - \frac{\partial Ploss}{\partial Pi} \right)$
prod_cost	Mendapatkan nilai biaya pembangkitan setiap unit setelah mendapatkan nilai pembangkitan yang optimal dari proses iterasi lambda
PROSES	Pada prosedur ini memproses <i>dynamic programming</i> hingga menghasilkan keluaran penjadwalan pembangkit dan juga pembebanannya.
FINAL_OUTPUT	Prosedur yang digunakan untuk mencetak hasil keluaran (<i>output</i>).

3.5 Iterasi Lambda pada Economic Dispatch

Perhitungan *economic dispatch* menggunakan iterasi lambda. *Economic dispatch* ini digunakan untuk mencari pembebanan optimal tiap kombinasi pada interval waktu tertentu di dalam program *unit commitment*. Tidak hanya digunakan pada *unit commitment*, *economic dispatch* juga digunakan untuk memeriksa hasil dari *unit commitment* apakah sudah memenuhi batasan *ramp-rate* atau belum.



Gambar 3.3 Flowchart iterasi lambda

Iterasi lambda diawali dengan memulai menentukan λ_{start} dengan rumus :

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2} \quad (3.1)$$

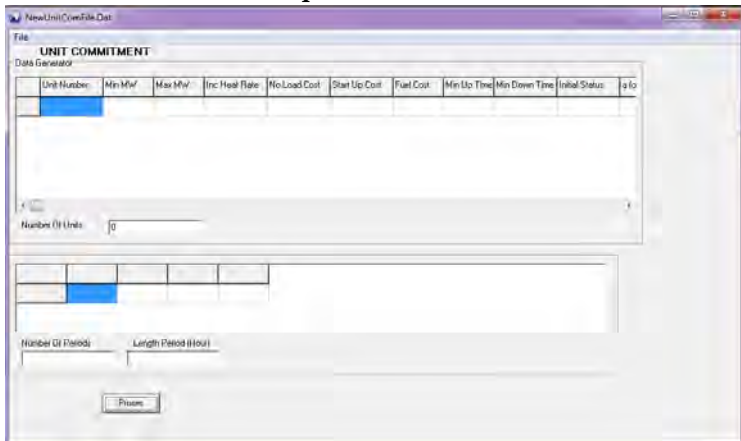
$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} \quad (3.2)$$

Nilai dari dari λ_{max} dapat ditentukan dengan menggunakan persamaan (2.12) dan nilai λ_{min} dapat ditentukan dengan menggunakan persamaan (2.11). Maka nilai lambda yang dapat digunakan dalam proses iterasi adalah

$$\lambda = \frac{\Delta\lambda}{2} \quad (2.23)$$

Lambda akan diiterasi hingga didapatkan nilai lambda yang sesuai. Nilai lambda akan berhenti saat *error* (ε) kurang dari *error* toleransi. Pada Tugas Akhir ini digunakan toleransi hingga 0.01. Selain itu perlu juga terpenuhinya syarat daya pembangkitan seperti pada persamaan (2.3) dan persamaan (2.7)

3.6 Penggunaan Software Perhitungan Unit Commitment and Economic Dispatch



Gambar 3.4 Menu utama unit commitment and economic dispatch

Gambar 3.5 adalah tampilan utama dari software perhitungan Unit commitment and Economic Dispatch. Number of unit merupakan tempat untuk mengisi jumlah unit yang akan dimasukkan. Untuk

memasukkan data *input*, diperlukan klik dua kali, kemudian akan muncul *window*. *Window* tersebut sebagai tempat untuk memasukkan data *input* unit pembangkit yang direpresentasikan pada Gambar3.6.

The 'Edit' window contains the following fields and values:

- Unit Number: 1
- Min (MW): 0.0
- Max (MW): 0.0
- Inc Heat Rate: 0.0
- No Load Cost: 0.0
- StartUpCost: 0.0
- Fuel Cost: 0.0
- Min Up Time: 0
- Min Down Time: 0
- Initial Status: 0
- a (orde 2): 0.000000
- b (orde 1): 0.000000
- c (orde 0): 0.000000
- dr: 0.00
- ur: 0.00
- P Sebelumnya: 0.00

Buttons: Ok, Cancel

Gambar 3.5 Editor input

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5	Periode 6	Periode 7	Periode 8	Periode 9	Periode 10	Periode

Number Of Periods: 24 Length Period (Hour): 1

Gambar 3.6 Input untuk beban setiap interval

Pada Gambar 3.7 merepresenntasikan tempat untuk memasukkan data beban setiap interval waktu yang ada. Data beban ini memiliki batas maksimal yaitu 24 interval.

Jika data *input* pembangkit dan data beban sudah diisikan, maka langkah selanjutnya yaitu klik proses. Setelah klik proses, akan muncul *window* baru sebagai berikut:



The screenshot shows a Windows-style application window titled "FmUCProses". It contains several input fields and radio button groups. The first field is labeled "Number Of Search States In Previous Period (N)" and is empty. The second field is labeled "Number Of Search States In Present Period (X)" and is also empty. Below these are two groups of radio buttons. The first group is labeled "Include Up/Down Time Constraints" and has two options: "Yes" (selected) and "No". The second group is labeled "Select Unit Order Option" and has one visible option: "Complete Enumeration" (selected). At the bottom center of the window is a button labeled "Proses".

Gambar 3.7 Editor untuk proses

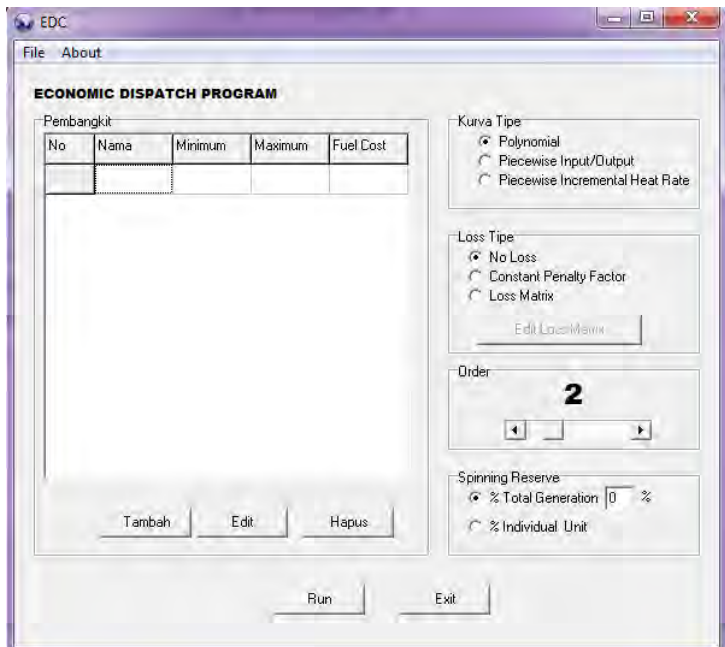
Pada Gambar 3.8, terdapat kotak untuk memasukkan nilai N dan X. N adalah jumlah *state* pada periode sebelumnya yang ingin dihitung. Sedangkan X adalah jumlah *state* untuk periode selanjutnya yang ingin dihitung.

Unitcom Result													
PERIOD 8 SYSTEM LOAD = 1483.000 MW													
STATE	UNIT STATUS										PCOST	PATH	FCOST
	1	2	3	4	5	6	7	8	9	10	R/HR		R
1	2	4	2	5	6	6	8	8	8	7	3412	1	18290.80
65	2	4	2	-1	6	6	8	8	8	7	3459	1	18337.70
513	-1	4	2	5	6	6	8	8	8	7	3435	1	18314.35
OPTIMUM COMMITMENT SCHEDULE													
TOTAL COST = 18290.88 R													
PERIOD	STATE	UNIT STATUS										PCOST	LOAD
		1	2	3	4	5	6	7	8	9	10	R/HR	MW
8	1	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	3412	1483.000
7	1	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	3175	1400.000
6	641	OFF	ON	OFF	ON	ON	ON	ON	ON	ON	ON	2769	1250.000
5	641	OFF	ON	OFF	ON	ON	ON	ON	ON	ON	ON	2610	1191.000
4	897	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON	ON	2130	1000.000
3	961	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON	1961	930.000
2	1009	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON	1286	641.000
1	1010	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	OFF	947	480.000
<div>Save Report</div> <div>Close</div> <div>Edc</div>													

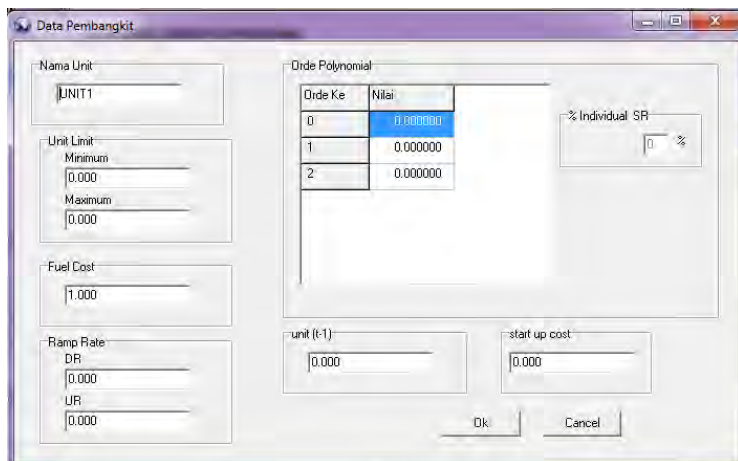
Gambar 3.8 Tampilan untuk hasil perhitungan

Pada Gambar 3.9 merupakan tampilan hasil perhitungan *unit commitment*. Pada hasil *unit commitment*, terdapat unit status setiap unit pembangkit untuk setiap interval. Selain itu ditampilkan pula besar daya pembangkitan dan juga biaya pada masing-masing interval. *Total cost* juga ditampilkan pada Gambar 3.9. *Total cost* didapat dari penjumlahan biaya pada setiap interval dan ditambah biaya *start-up cost*.

Untuk memeriksa hasil *ramp-rate*, klik EDC pada window tersebut. Berikut ini adalah tampilan untuk *software economic dispatch*:



Gambar 3.9 Main menu EDC



Gambar 3.10 Input Editor untuk EDC

Set up Solution

Solution Method

☒ Lambda Search

☐ Table Look Up

Maximum generation is : 910.0

Minimum generation is : 300.0

Spinning Reserve is : 0.0

Ramp-rate

☒ Yes ☐ No

Schedule Type

☒ Total Generation

☐ Total Load

Enter Total Generation

Ok

Gambar 3.11 Tampilan untuk memilih *ramp-rate* dan *input* beban

Result

```

lambda = 15.7176 totalgen = 485.0
lambda = 15.7081 totalgen = 485.0
lambda = 15.7033 totalgen = 485.0
lambda = 15.7009 totalgen = 485.0
lambda = 15.6997 totalgen = 485.0
lambda = 15.6991 totalgen = 485.0
lambda = 15.6988 totalgen = 485.0
lambda = 15.6986 totalgen = 485.0
lambda = 15.6986 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0
lambda = 15.6985 totalgen = 485.0

```

generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	335.0	min	16.4580	1.0000	6468.54
UNIT2	150.0	min	17.3500	1.0000	3565.75
totals	485.0				10034.29

lambda = 15.6985
start up cost = 0.0000
final total cost = 10034.2900

Save Report Close

Gambar 3.12 Hasil EDC

Pada Gambar 3.10 tampilan untuk menu utama dari *software economic dispatch*. Pada bagian ini perlu diklik bagian tambah untuk menambah unit pembangkit. Untuk memasukkan dari *input* pembangkit, klik bagian *edit* kemudian akan muncul Gambar 3.11. Setelah data *input* selesai dimasukkan, klik bagian *run* untuk menuju tahapan seperti Gambar 3.12. Klik bagian *ramp-rate* Yes untuk mempertimbangkan batasan *ramp-rate*. Kemudian masukkan nilai beban pada kotak *enter total generation*. Setelah itu klik *ok* dan akan keluar hasilnya seperti pada Gambar 3.13.

Halaman ini sengaja dikosongkan

BAB 4

SIMULASI DAN ANALISIS

Pada bab ini akan diperlihatkan hasil perhitungan *unit commitment* dengan menggunakan metode *dynamic programming*. Hasil perhitungan yang diperlihatkan berupa tabel-tabel yang meliputi penjadwalan pembangkit, pembangkitan optimal pembangkit yang *on*, dan biaya total dengan berbagai kondisi beban. Hasil simulasi ini akan menampilkan beberapa studi kasus. Dalam Tugas Akhir ini, data yang dimasukkan merupakan asumsi dari referensi jurnal.

4.1 Studi Kasus 1

Studi kasus 1 memperlihatkan hasil perhitungan ketika data masukan menggunakan *input* linier. Program yang digunakan adalah program yang belum dikembangkan pada Tugas Akhir ini. Berikut ini adalah data yang dimasukkan dalam simulasi:

Tabel 4.1 Data *input* studi kasus 1

Keterangan	Unit Pembangkit			
	1	2	3	4
Pmax (MW)	80	250	300	60
Pmin (MW)	25	60	75	20
Incremental Heat Rate (Btu/KWh)	10.440	9.000	8.730	1.190 0
No Load Cost (R/h)	213	585,62	684,74	252
Start Up Cost (R/mWh)	350	400	1.100	0
Fuel Cost	2	2	2	1,2
MUT	4	5	5	1
MDT	2	3	4	1
Initial status	-5	8	8	-6

Dengan bebannya sebagai berikut:

Tabel 4.2 Data beban studi kasus 1

Interval ke-	Beban (MW)
1	450
2	530
3	600
4	540
5	400
6	280
7	290
8	500

Berdasarkan data di atas, maka didapatkan hasil simulasi sebagai berikut:

Tabel 4.3 Hasil simulasi studi kasus 1

Interval ke-	Unit status				Pcost (\$/hr)
	1	2	3	4	
1	OFF	ON	ON	OFF	9.208
2	OFF	ON	ON	OFF	10.648
3	OFF	ON	ON	ON	11.974
4	OFF	ON	ON	OFF	10.828
5	OFF	ON	ON	OFF	8.308
6	OFF	ON	ON	OFF	6.192
7	OFF	ON	ON	OFF	6.366
8	OFF	ON	ON	OFF	10.108

Total biaya pembangkitan sebesar 73.633,40 \$

Dari simulasi di atas, terlihat bahwa unit pembangkit 2 dan 3 selalu terhubung dengan sistem (ON). Pada interval ke-3 dengan beban 600 MW, unit 2 dan 3 tidak mampu untuk memenuhi permintaan beban tersebut sehingga unit 4 diaktifkan.

Pada program *unit commitment* ini hanya menampilkan unit yang ON dan OFF saja, namun tidak dapat menampilkan berapa besar pembangkitan setiap pembangkitnya. Oleh karena itu pada studi kasus 2 dan 3 akan terlihat pembangkitan setiap unit pembangkit pada periode tertentu berdasarkan software yang telah dikembangkan.

4.2 Studi Kasus 2

Pada studi kasus kedua ini memperlihatkan hasil perhitungan ketika data masukan menggunakan *input* nonlinier. Program yang digunakan adalah program yang telah dikembangkan untuk Tugas Akhir ini.

Data masukan yang digunakan adalah sistem 10 unit pembangkit dengan 24 interval pembebanan [5]. Berikut ini adalah data masukannya:

Tabel 4.4 Data *input* studi kasus 2

Keterangan	Unit Pembangkit				
	1	2	3	4	5
Pmax	455	455	130	130	162
Pmin	150	150	20	20	25
a	0,0004	0,0003	0,002	0,0021	0,0039
b	16,19	17,26	16,6	16,5	19,7
c	1000	970	700	680	450
MUT	8	8	5	5	6
MDT	8	8	5	5	6
UR	100	100	130	130	162
DR	120	120	130	130	162
Start up cost	4500	5000	550	560	900
Initial status	8	8	-5	-5	-6
Keterangan	Unit Pembangkit				
	6	7	8	9	10
Pmax	80	85	55	55	55
Pmin	20	25	10	10	10
a	0,0071	0,0079	0,0022	0,0017	0,0041
b	22,36	27,74	27,27	27,79	25,92
c	370	480	665	670	660
MUT	3	3	1	1	1
MDT	3	3	1	1	1
UR	80	85	55	55	55
DR	80	85	55	55	55
Start up cost	170	260	30	30	30
Initial status	-3	-3	-1	-1	-1

Dengan pembebanan sebagai berikut:

Tabel 4.5 Data beban studi kasus 2

Interval ke-	Beban (MW)
1	700
2	750
3	850
4	950
5	1.000
6	1.100
7	1.150
8	1.200
9	1.300
10	1.400
11	1.450
12	1.500
13	1.400
14	1.300
15	1.200
16	1.050
17	1.000
18	1.100
19	1.200
20	1.400
21	1.300
22	1.100
23	900
24	800

Berdasarkan data pada tabel 4.7, maka didapatkan hasil simulasi sebagai berikut:

Tabel 4.6 Hasil simulasi studi kasus 2

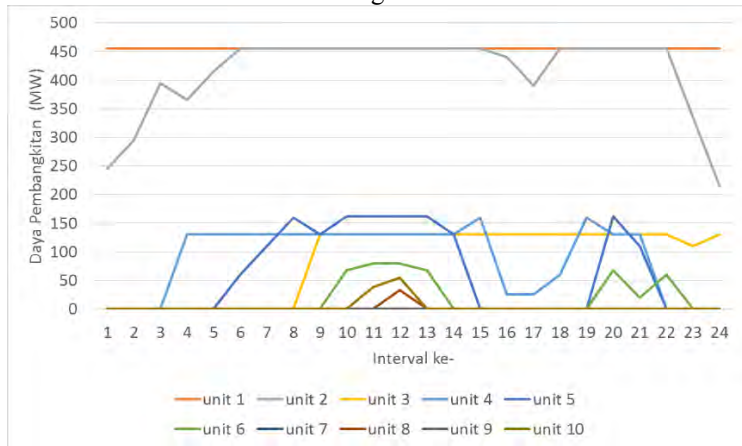
Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	455	245	0	0	0	0
2	455	295	0	0	0	0
3	455	395	0	0	0	0
4	455	365	0	130	0	0
5	455	415	0	130	0	0
6	455	455	0	130	60	0
7	455	455	0	130	110	0
8	455	455	0	130	160	0
9	455	455	130	130	130	0
10	455	455	130	130	162	68
11	455	455	130	130	162	80
12	455	455	130	130	162	80
13	455	455	130	130	162	68
14	455	455	130	130	130	0
15	455	455	130	160	0	0
16	455	440	130	25	0	0
17	455	390	130	25	0	0
18	455	455	130	60	0	0
19	455	455	130	160	0	0
20	455	455	130	130	162	68
21	455	455	130	130	110	20
22	455	455	130	0	0	60
23	455	335	110	0	0	0
24	455	215	130	0	0	0

Tabel 4.6 Hasil simulasi studi kasus 2 (lanjutan)

Interval ke-	Daya Pembangkitan (MW)				
	unit 7	unit 8	unit 9	unit 10	cost per interval (\$)
1	0	0	0	0	13.665,9
2	0	0	0	0	14.537,22
3	0	0	0	0	16.283,77
4	0	0	0	0	18.619,7
5	0	0	0	0	19.494,36
6	0	0	0	0	21.841,26
7	0	0	0	0	22.859,35
8	0	0	0	0	23.896,94
9	0	0	0	0	26.163,69
10	0	0	0	0	28.754,03
11	0	0	0	38	30.686
12	0	33	0	55	32.700
13	0	0	0	0	28.754
14	0	0	0	0	26.164
15	0	0	0	0	23.897
16	0	0	0	0	20.877,23
17	0	0	0	0	20.001,75
18	0	0	0	0	21.841,26
19	0	0	0	0	23.896,94
20	0	0	0	0	28.754,03
21	0	0	0	0	26.571,31
22	0	0	0	0	21.963,64
23	0	0	0	0	17.785,39
24	0	0	0	0	16.035,83

Dengan total biaya pembangkitan 548996,64 \$

Jika direpresentasikan dalam bentuk grafik, maka hasil simulasi studi kasus 2 adalah sebagai berikut:



Gambar 4.1 Grafik pembebanan studi kasus 2

Berdasarkan gambar 4.1, terlihat bahwa unit 1 selalu *commit (on)* dengan pembangkitan maksimal. Unit 2 menjadi pembangkit kedua yang pembangkitannya terbesar. Hal ini karena unit 1 dan unit 2 memiliki fungsi biaya termurah diantara pembangkit yang lain.

Pada studi kasus 2 ini efek *ramp-rate* tidak terlalu terlihat. Meskipun pada data *input* terdapat UR dan DR sebagai batasan *ramp-rate*, namun UR dan DR tersebut memiliki nilai yang besar. Besarnya batasan *ramp-rate* tersebut menyebabkan perubahan besar pembangkitan pada unit yang aktif tidak selalu masuk dalam batasan *ramp-rate*.

Efek dari batasan *minimum up-time* dan *minimum down-time* juga berpengaruh pada hasil simulasi studi kasus 2. Unit 3 mulai *commit* saat memasuki interval ketiga karena batasan *minimum down-time* sudah terpenuhi saat interval ketiga.

4.3 Studi Kasus 3

Pada studi kasus ketiga ini memperlihatkan hasil perhitungan ketika data masukan menggunakan *input* nonlinier dengan memperhatikan batasan *ramp-rate*. Program yang digunakan adalah program yang telah dikembangkan untuk Tugas Akhir ini.

Data masukan yang digunakan adalah sistem 10 unit pembangkit [6]. Data masukan menggunakan 8 interval pembebanan. Berikut ini adalah data masukannya:

Tabel 4.7 Data *input* studi kasus 3

Keterangan	Unit Pembangkit				
	1	2	3	4	5
Pmax (MW)	73	93	143	70	93
Pmin (MW)	12	26	42	18	30
a	0,0051	0,0040	0,0039	0,0038	0,0021
b	2,2034	1,9101	1,8518	1,6966	1,8015
c	15	25	40	32	29
UR	10	15	23	10	13
DR	10	15	23	10	13
Keterangan	Unit Pembangkit				
	6	7	8	9	10
Pmax	350	248	190	190	113
Pmin	100	100	40	70	40
a	0,0026	0,0029	0,0015	0,0013	0,0014
b	1,5354	1,2643	1,2130	1,1954	1,1285
c	72	49	82	105	100
UR	50	48	20	30	20
DR	50	48	20	30	20

Pada data masukan minimum up time, minimum down time, start up cost, dan initial condition diabaikan atau dianggap nol. Namun efek *ramp-rate* diperhitungkan dengan nilai UR dan DR yang relatif kecil, tidak seperti studi kasus 2 yang UR dan DR memiliki nilai yang besar. Untuk nilai pembebanannya adalah sebagai berikut:

Tabel 4.8 Data beban studi kasus 3

Interval ke-	Beban (MW)	Interval ke-	Beban (MW)
1	480	5	1.191
2	641	6	1.250
3	930	7	1.400
4	1.000	8	1.483

Berdasarkan data *input*, maka didapatkan hasil simulasi sebagai berikut:

Tabel 4.9 Hasil simulasi studi kasus 3

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	163,20
4	0	0	0	70	93	163,20
5	0	90,52	0	70	93	216,17
6	0	93	0	70	93	253,01
7	53,85	93	108,88	70	93	240,25
8	73	93	134,43	70	93	278,58
Interval ke-	Daya Pembangkitan (MW)					cost per interval (\$)
	unit 7	unit 8	unit 9	unit 10		
1	100	190	190	0	947	
2	147,99	190	190	113	1.286	
3	180,80	190	190	113	1.961	
4	180,80	190	190	113	2.130	
5	228,30	190	190	113	2.610	
6	248	190	190	113	2.769	
7	248	190	190	113	3.175	
8	248	190	190	113	3.412	
Total biaya pembangkitan					18.290	

Berdasarkan hasil simulasi, beberapa unit pembangkit melangar batasan *ramp-rate*.

Tabel 4.10 Interval yang masih melanggar ramp-rate

Unit	interval	Kenaikan/ Penurunan daya (MW)	UR/DR
1	7 ke 8	+19.14518	10
3	7 ke 8	+25.54273	23
6	4 ke 5	+52.97414	50

Oleh karena pada hasil *unit commitment* terdapat unit pembangkit yang belum memenuhi *ramp-rate*, maka diperlukan perhitungan kembali dengan program *economic dispatch* dengan memperhatikan *ramp-rate*.

Dengan data masukan yang sama, program *economic dispatch* digunakan untuk mencari pembebanan optimal dengan memperhitungkan *ramp-rate*. Data dihitung setiap interval. Sedangkan *ramp-rate* diasumsikan digunakan pada setiap perubahan pembangkitan unit pembangkit yang ON. Untuk perubahan dari pembangkit yang OFF menjadi ON diasumsikan tidak menggunakan batasan *ramp-rate*.

Berikut ini adalah hasil simulasi setelah dikoreksi program *economic dispatch* dengan memperhatikan *ramp-rate*:

Tabel 4.11 Hasil simulasi studi kasus 3 dengan ramp-rate

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	156,7
4	0	0	0	70	93	156,7
5	0	93	0	70	93	206,7
6	0	93	0	70	93	253
7	53,7	93	115,4	70	93	233,9
8	63,7	93	1138,4	70	93	283,9

Tabel 4.11 Hasil simulasi studi kasus 3 dengan ramp-rate (lanjutan)

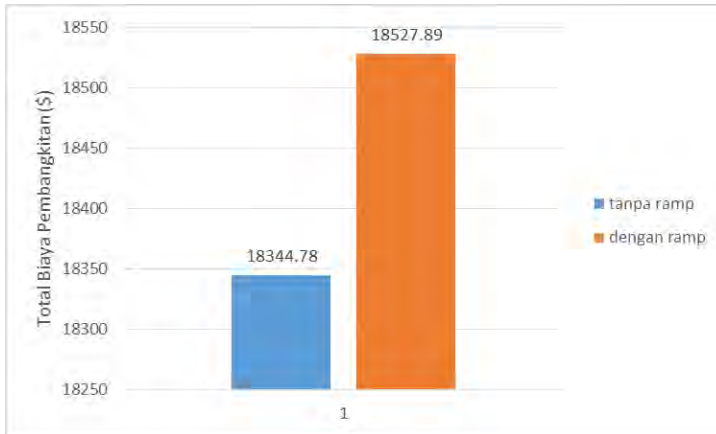
Interval ke-	Daya Pembangkitan (MW)				
	unit 7	unit 8	unit 9	unit 10	cost per interval (\$)
1	100	190	190	0	947
2	148	190	190	113	1.290,71
3	187,3	190	190	113	1.969,75
4	187,3	190	190	113	2.139,13
5	235,3	190	190	113	2.619,9
6	248	190	190	113	2.780,17
7	248	190	190	113	3.180,32
8	248	190	190	113	3.417,8
Total biaya pembangkitan					18.344,78

Berdasarkan hasil di atas, pembangkitan unit pembangkit sudah memenuhi *ramp-rate*. Tabel 4.15 merepresentasikan *ramp-rate* yang sudah terpenuhi.

Tabel 4.12 Interval yang sudah tidak melanggar ramp-rate

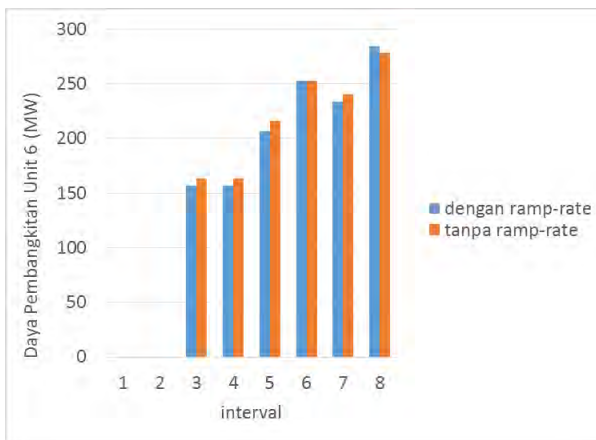
Unit	interval	Kenaikan/ Penurunan daya (MW)	UR/DR
1	7 ke 8	+10	10
3	7 ke 8	+23	23
6	4 ke 5	+50	50

Jika hasil simulasi yang pertama dan kedua dibandingkan, maka hasil simulasi yang kedua memiliki total biaya pembangkitan yang lebih tinggi. Hal ini karena hasil simulasi yang kedua memperhatikan batasan *ramp-rate*, sehingga perhitungan pembangkitan pada setiap semakin ketat dan kompleks.



Gambar 4.2 Total biaya perhitungan dengan ramp-rate dan tidak dengan ramp-rate

Dari Gambar 4.2 terlihat bahwa ketika ramp-rate diperhitungkan akan memiliki biaya yang lebih tinggi daripada perhitungan yang mengabaikan ramp-rate. Perbedaan harganya adalah sebesar 183.11 \$. Hal ini dikarenakan batasan ramp-rate menyebabkan variasi daya pembangkitan yang lebih ketat.



Gambar 4.3 Pembangkitan unit 6

Gambar 4.3 adalah grafik pembangkitan pada unit 6. Untuk perhitungan tanpa ramp-rate (kolom oranye), dari interval 4 ke interval 5 naik melebihi batas *ramp-rate*. Sedangkan untuk perhitungan dengan ramp-rate (kolom biru), saat dari interval 4 ke interval 5 naik tidak melebihi batas *ramp-rate*.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil simulasi dan analisis diperoleh kesimpulan sebagai berikut:

1. Aplikasi perhitungan *unit commitment* dapat melakukan perhitungan sesuai batasan minimum up time dan minimum down time dengan dengan *input* nonlinier. Pada studi kasus 2 hasil *output* perhitungan didapatkan total biaya pembangkitan 548.996,64 \$.
2. Diperlukan bantuan aplikasi perhitungan *economic dispatch* jika hasil keluaran dari *unit commitment* melangar *ramp-rate*. Aplikasi perhitungan *economic dispatch* sudah dapat menghitung pembebanan optimal pada setiap pembangkit dengan mempertimbangkan *ramp-rate*.
3. Total biaya pembangkitan dengan mempertimbangkan *ramp-rate* lebih besar daripada tanpa batasan *ramp-rate* karena batasan *ramp-rate* menyebabkan variasi daya pembangkitan yang lebih ketat. Pada hasil studi kasus 3, total biaya pembangkitan dengan mempertimbangkan *ramp-rate* lebih mahal 183,11 \$.

5.2 Saran

Adapun saran untuk penelitian selanjutnya yang berkaitan dengan Tugas Akhir ini, yaitu:

1. Software ini dapat ditambahkan spinning reserve.
2. Permasalahan dapat dikembangkan dengan menambahkan rugi saluran transmisi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Wang, C., dan Shahidehpour, S. M., “*Effect of Ramp-Rate Limits on Unit Commitment and Economic Dispatch*”, IEEE Paper, 1993.
- [2] Saadat, Hadi. “*Power System Analysis 2nd Edition*”, McGrawHill, Ch.1, 1999.
- [3] Wood, A.J., dan Bruce F.W., “*Power Generation, Operation, and Control*”, A Wiley-Interscience Publication. New York, 1984.
- [4] Kunnassriyanto, “Belajar Pemrograman Delphi”, MODULA, Bandung, 2011.
- [5] Sundaram, C. S., Sudhakaran, M., Selvakumar, R., Natarajan, R. G., “*Solution to Profit Based Unit Commitment using Swarm Intelligence Technique*”, Proc. of ICCPEIC, 2014.
- [6] Abidin, Zainal, “*Dynamic Economic Dispatch Menggunakan Quadratic Programming*”, Buku Tugas Akhir, Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2012.

Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Penulis memiliki nama lengkap Wildan Nurwanto. Penulis lahir di Serang pada tanggal 8 Oktober 1992. Penulis merupakan anak kedua dari tiga bersaudara pasangan Wawan Gunawan dan Ani Umiyati. Penulis memulai pendidikan di SDN Serayu Yogyakarta. Setelah itu penulis melanjutkan pendidikan di SMPN 5 Yogyakarta. Setelah lulus dari SMPN 5 Yogyakarta pada tahun 2008, penulis melanjutkan sekolah di SMAN 1 Yogyakarta.

Pada tahun 2011, penulis lulus dari SMAN 1 Yogyakarta kemudian melanjutkan pendidikan di Jurusan Teknik Elektro, FTI, ITS. Penulis juga aktif dalam berbagai kepanitiaan dan organisasi. Penulis pernah menjadi panitia Elektra dan pernah aktif menjadi anggota di Divisi Workshop Elektro ITS. Penulis dapat dihubungi melalui alamat e-mail: wildanurwanto@gmail.com

Halaman ini sengaja dikosongkan

Pengembangan *Software Unit Commitment and Economic Dispatch* dengan Mempertimbangkan *Ramp-rate*

Wildan Nurwanto, Ontoseno Penangsang, dan Arif Musthofa

Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

e-mail: wildanurwanto@gmail.com, ontosenop@ee.its.ac.id, arif@ee.its.ac.id

Abstrak--Kemajuan teknologi dan ekonomi membuat permintaan beban listrik meningkat. Meningkatnya beban listrik harus diimbangi dengan penambahan daya yang dibangkitkan. Hal ini menjadikan penjadwalan unit pembangkit perlu ditentukan dengan baik agar didapatkan pembangkitan dengan harga yang optimal. Tugas Akhir ini mengembangkan *software* Powergen yang digunakan dalam perhitungan *unit commitment* dan *economic dispatch* namun dengan mempertimbangkan *ramp-rate*. Metode yang digunakan adalah *complete enumeration* dengan *dynamic programming* pada *unit commitment*. Sedangkan pada *economic dispatch* menggunakan metode iterasi lambda. Metode-metode tersebut diterapkan pada Delphi sehingga diharapkan akan menghasilkan *software* perhitungan *unit commitment* dan *economic dispatch* dengan mempertimbangkan *ramp-rate*.

Kata Kunci—Delphi, *economic dispatch*, *ramp-rate*, *unit commitment*.

I PENDAHULUAN

Pada sistem tenaga listrik, pengaturan penjadwalan unit pembangkit dibutuhkan untuk mendapatkan biaya pembangkitan yang paling ekonomis. Permintaan beban yang terus berubah setiap jam, hari, minggu, bulan, bahkan tahun mengakibatkan perubahan besar daya yang disalurkan pada sisi pembangkitan. Hal ini menyebabkan terjadinya mekanisme penyambungan dan pelepasan unit pembangkit pada sistem kelistrikan yang disebut *unit commitment*. Untuk mendapatkan biaya pembangkitan minimal maka digunakan juga *economic dispatch*.

Pada saat ini aplikasi perhitungan *unit commitment* dan *economic dispatch* masih sangat terbatas. Dosen maupun mahasiswa Teknik Elektro ITS biasanya memakai aplikasi PowerGen untuk perhitungan tersebut. Namun PowerGen belum memiliki parameter *ramp-rate* untuk perhitungan *unit commitment* dan *economic dispatch*. Selain itu, program *unit commitment* yang sudah ada hanya dapat memasukkan *input* yang memiliki fungsi linier. Oleh karena itu pada Tugas Akhir kali ini akan mengembangkan *software* tersebut dengan menambahkan *input* nonlinier pada *unit commitment* dan menghitung pembangkitan dalam *unit commitment* tersebut diselesaikan dengan *economic dispatch*. Setelah itu diperiksa kembali apakah hasil yang sudah didapat melebihi batasan *ramp-rate* atau tidak.

II UNIT COMMITMENT AND ECONOMIC DISPATCH

A. Economic Dispatch

Economic dispatch adalah suatu pembagian pembebanan pada unit-unit pembangkit secara optimal ekonomi yang ada

dalam sistem, pada harga beban sistem tertentu. Oleh karena beban sistem yang selalu berubah dari waktu ke waktu, maka untuk perhitungan *economic dispatch* tersebut juga dilakukan setiap beban tersebut.

Optimasi yang dilakukan pada permasalahan *economic dispatch* merupakan optimasi dari sisi biaya bahan baku pembangkitan atau *fuel cost*. Biaya bahan baku pembangkitan ini memiliki karakteristik nonlinier. Bentuk dari persamaan biaya pembangkitan adalah persamaan *polynomial* orde dua yang dapat dituliskan sebagai berikut:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (1)$$

keterangan:

F_i = biaya pembangkitan pada pembangkitan ke-i (\$)

P_i = Daya output unit pembangkit i (MW)

Variabel a , b , c merupakan konstanta biaya operasi dari suatu pembangkit. Koefisien c menunjukkan biaya operasi pembangkit ketika tidak memproduksi energi listrik.

Parameter-parameter *equality constraint* dan *inequality constraint* dapat direpresentasikan dalam persamaan berikut

$$\text{Min } \Sigma F_i(P_i) = \text{Min } \Sigma a_i P_i^2 + b_i P_i + c_i \quad (2)$$

$$P_{Gi} \text{min} \leq P_G \leq P_{Gi} \text{max} \quad (3)$$

P_{Gi} adalah besarnya daya yang dibangkitkan generator ke-i atau persamaan (3) ini disebut dengan *inequality constraint*.

$$\Sigma P_i = P_d + P_L \quad (4)$$

Keterangan:

P_d : daya permintaan konsumen (MW)

P_L : rugi daya yang terjadi pada jaringan transmisi (MW)

Persamaan (4) disebut dengan *equality constraint*. *Transmission losses* atau P_L pada persamaan (4) dipengaruhi oleh besarnya aliran daya yang mengalir pada jaringan transmisi.

Ramp-rate adalah kemampuan generator untuk melakukan peningkatan (*up-rate*) atau penurunan (*down-rate*) pembangkitan. Setiap unit pembangkit memiliki karakteristik *ramp-rate* yang berbeda. Oleh karena itu perhitungan *ramp-rate* membutuhkan fungsi tertentu untuk hasil yang optimal.

Fungsi *ramp-rate* dapat direpresentasikan dalam persamaan sebagai berikut:

$$P_i(t-1) - P_i(t) \leq DR_i \quad (5)$$

$$P_i(t-1) - P_i(t) \leq UR_i \quad (6)$$

Jika persamaan (5) dan persamaan (6) disatukan, maka akan mendapatkan batasan baru, yaitu:

$$P_i(t-1) - DR_i \leq P_i(t) \leq DR_i + P_i(t-1) \quad (7)$$

Keterangan:

$P_i(t)$: pembangkitan unit i pada interval ke-t (MW)

$P_i(t-1)$: pembangkitan unit i pada interval ke t-1 (MW)

B. Unit Commitment

Kebutuhan pemakaian energi (beban) listrik mengikuti siklus kegiatan manusia sehari-hari. Naik atau turunnya pemakaian energi listrik selalu diimbangi oleh pembangkitan energi listrik dalam sistem. Untuk mengikuti siklus pembangkitan energi listrik tersebut dilakukan penjadwalan unit yang *commit (on)* dan unit yang *off* dalam siklus waktu tertentu. Penjadwalan itulah yang disebut dengan *unit commitment*.

Pada *unit commitment* dikenal dengan kombinasi unit pembangkit. Kombinasi unit pembangkit merupakan kombinasi *on/off* dari beberapa unit yang ada dalam sistem. Jika ada N pembangkit, maka dapat dihitung jumlah kombinasi yang mungkin adalah $2^N - 1$. Dari kombinasi yang ada akan dipilih kombinasi yang dijadwalkan. Evaluasi pemilihan kombinasi yang masuk dalam jadwal pembangkitan dilakukan dengan menghitung biaya optimum (*economic dispatch*) untuk setiap kombinasi sehingga dapat dibandingkan biaya optimum terendah dari kombinasi-kombinasi *on/off* unit pembangkit pada beban tertentu.

Pada *unit commitment* ada beberapa *constraint* yang dipertimbangkan, diantaranya *minimum up-time*, *minimum down-time*, *start-up cost*.

C. Iterasi Lambda

Untuk nilai lambda terdapat pada persamaan lagrangian:

$$\frac{\partial Fi}{\partial Pi} = \lambda \left(1 - \frac{\partial Ploss}{\partial Pi} \right) \quad (8)$$

$$\lambda = \frac{\partial Fi}{\partial Pi} \left(\frac{1}{1 - \frac{\partial Ploss}{\partial Pi}} \right) \quad (9)$$

$$\lambda = \frac{\partial Fi}{\partial Pi} Pfi \quad (10)$$

Untuk menentukan nilai lambda awal direpresentasikan pada persamaan (13):

$$\lambda_{min} = \min \left(\frac{\partial Fi}{\partial Pi} Pfi, i = 1 \dots \text{Jumlah Generator} \right) \quad (11)$$

$$\lambda_{max} = \max \left(\frac{\partial Fi}{\partial Pi} Pfi, i = 1 \dots \text{Jumlah Generator} \right) \quad (12)$$

$$\lambda_{start} = \frac{\lambda_{max} - \lambda_{min}}{2} \quad (13)$$

Untuk melakukan iterasi maka diperlukan $\Delta\lambda$, yang dapat direpresentasikan pada persamaan (14)

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} \quad (14)$$

$$P_{target} = P_{loss} + P_{load} \quad (15)$$

Jika nilai yang didapatkan dalam proses iterasi sama dalam persamaan (16), maka solusi perubahan nilai lambda untuk iterasi berikutnya adalah (17).

$$\sum_i^n Pi - P_{target} > 0, \quad (16)$$

$$n = \text{Jumlah Generator}$$

$$\lambda = \lambda_{sebelum} - \Delta\lambda \quad (17)$$

Jika nilai yang didapatkan dalam proses iterasi sama dalam persamaan (18), maka solusi perubahan nilai lambda untuk iterasi berikutnya adalah (19).

$$\sum_i^n Pi - P_{target} < 0, \quad (18)$$

$$n = \text{Jumlah Generator}$$

$$\lambda = \lambda_{sebelum} + \Delta\lambda \quad (19)$$

Nilai lambda akan terus berubah hingga mendapatkan nilai seperti pada persamaan (2.20).

$$\sum_i^n Pi - P_{target} = 0, \quad (20)$$

$$n = \text{Jumlah Generator}$$

Nilai λ pada iterasi kedua bernilai 10 % lebih besar dari nilai λ pertama, atau 10 % kurang dari nilai λ pertama, bergantung pada hasil perhitungan *error* pada iterasi tertentu.

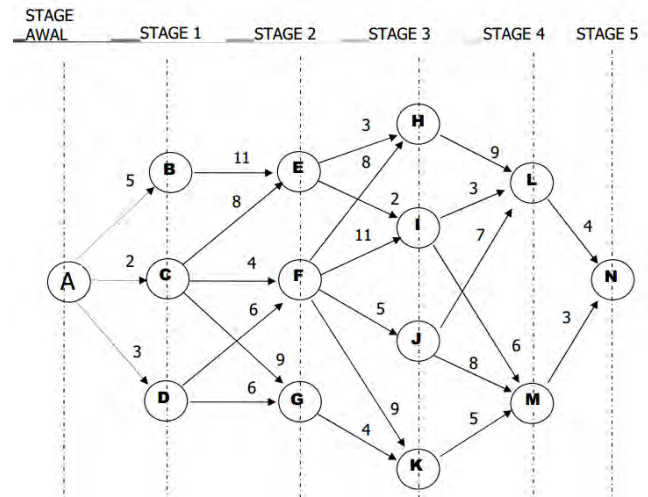
Pada Tugas Akhir ini nilai *error* (ϵ) ditentukan sebesar 0.01. Ketika diimplementasikan ke dalam persamaan, maka nilai lambda akan berhenti melakukan iterasi hingga mendapatkan nilai seperti pada persamaan (21)

$$\sum_i^n Pi - P_{target} = \epsilon, \quad (21)$$

$$n = \text{Jumlah Generator}$$

D. Dynamic Programming

Untuk menyelesaikan *unit commitment* digunakan *dynamic programming* dengan penelusuran ke depan (*forward approach*). Gambar 1 adalah contoh dari *dynamic programming*.



Gambar 1. Contoh dari *dynamic programming*

Keterangan:

State : terminal-terminal dengan kondisi tertentu.

Stage : kumpulan dari *state* pada level tertentu.

Gambar 1 menunjukkan model pembiayaan suatu proses yang diawali dari *state* A sampai ke *state* N dengan melalui banyaknya pilihan jalur pembiayaan. Terdapat lima *state*, setiap *stage* memiliki beberapa *state*.

Pada *dynamic programming*, persamaan dapat direpresentasikan sebagai berikut:

$$F_n(S, x_n) = C_{S,x_n} + F_{n-1}^*(x_{n-1}) \quad (22)$$

Keterangan:

x_n : *stage* ke- n

S : variabel *state* yang berada di *stage*

x_{n-1}

C_{S,x_n} : biaya dari *state* S ke *state* yang berada di x_n

$F_n(S, x_n)$: biaya kumulatif untuk sampai ke *state* yang berada di x_n melewati variable *state* S

$F_n^*(x_n)$: biaya kumulatif termurah untuk sampai ke *state* yang berada di x_n melewati variable *state* S

$F_{n-1}^*(x_{n-1})$: biaya kumulatif untuk sampai ke *state* yang berada di x_{n-1}

III IMPLEMENTASI DYNAMIC PROGRAMMING PADA UNIT COMMITMENT

Dynamic programming digunakan untuk menyelesaikan permasalahan dalam perhitungan *unit commitment*, sedangkan untuk pembebanan optimalnya digunakan iterasi lambda. Pengaruh dari batasan *ramp-rate* dikoreksi sesaat setelah pengecekan hasil akhir *unit commitment* dengan pembebanannya. Jika *ramp-rate* sudah terpenuhi, perhitungan sudah selesai. Jika *ramp-rate* belum terpenuhi, maka dilakukan perhitungan ulang dengan *economic dispatch* pada interval waktu tersebut.

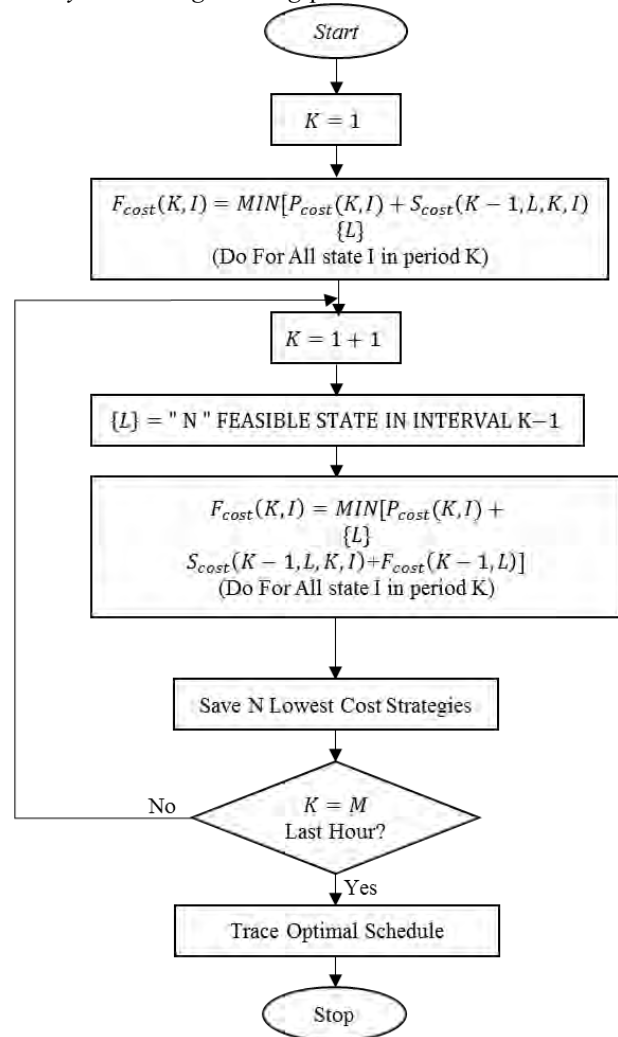
Pengolahan data dan simulasi menggunakan aplikasi perhitungan yang dibuat dengan menggunakan program Delphi.

Alur dimulai dengan menentukan fungsi objektif dan juga variabel *constraint*. Setelah itu program *unit commitment* dikembangkan sehingga dapat digunakan *input* nonlinier. Program *unit commitment* sebelumnya hanya dapat memasukkan data *input* unit pembangkit linier saja, sehingga saat ini sudah dapat digunakan data *input* unit pembangkit nonlinier.

Hasil dari program *unit commitment* ini adalah penjadwalan unit yang *commit (on)* dan juga unit yang *off*. Selain itu, hasil dari program *unit commitment* juga dapat menampilkan pembebanan unit *on* tersebut. Langkah selanjutnya adalah memastikan apakah pembebanan tersebut sudah sesuai dengan *ramp-rate* atau belum. Jika sudah maka hasilnya dapat dianalisis. Jika belum sesuai *ramp-rate*, maka langkah selanjutnya adalah menghitung kembali hasil simulasi dengan program *economic dispatch*. Program *economic dispatch* tersebut sudah dapat menghitung efek dari batasan *ramp-rate*. Perlu dilakukan pengecekan setiap interval waktu untuk memastikan apakah hasil dari *unit commitment* sudah memenuhi batasan *ramp-rate* atau belum.

Kemudian hasil akhir dari simulasi program dapat dianalisis dan disimpulkan sebagai kesimpulan dari Tugas Akhir ini.

A. *Dynamic Programming* pada *Unit commitment*



Gambar 2. Flowchart *unit commitment* dengan metode *dynamic programming*

Metode *dynamic programming* yang digunakan pada *unit commitment* dapat dilihat pada Gambar 2.

Keterangan:

State : berisikan kombinasi unit-unit pembangkit yang beroperasi dan tidak beroperasi.

Stage : level beban pada jam tertentu (interval waktu). Di dalam *stage* terdapat beberapa *state*.

N Feasible state : N buah *state* pada *stage* sebelumnya yang dipilih dengan biaya akumulatif termurah dimulai dari urutan yang termurah.

K : menyatakan *stage*/jam ke sekian untuk level-level beban tertentu

$F_{cost}(K, I)$: biaya kumulatif yang diperlukan untuk sampai ke *state* I pada level K

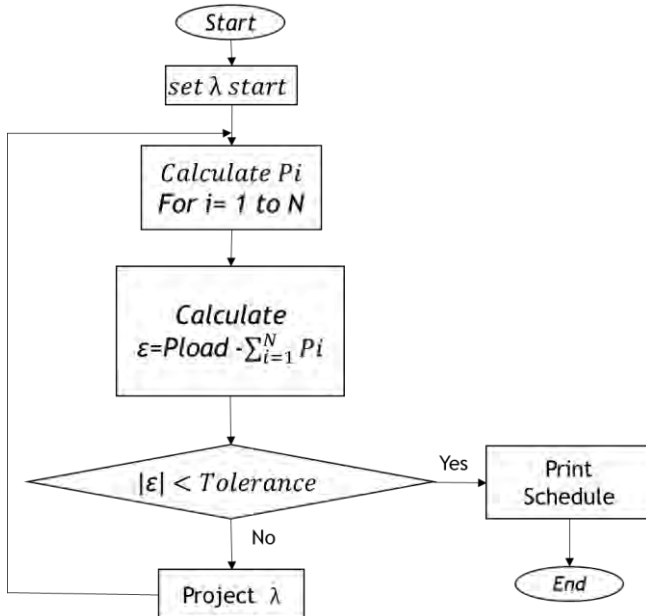
$F_{cost}(K, I)$: biaya produksi pada *state* I untuk level *stage* K

$S_{cost}(K - 1, L, K, I)$: biaya transisi dari *state* L pada level $K-1$ ke *state* I pada level K

X : jumlah *state* yang harus diperiksa pada setiap interval waktu (pada setiap *stage*)

B. Iterasi Lambda pada Economic Dispatch

Perhitungan *economic dispatch* menggunakan iterasi lambda. *Economic dispatch* ini digunakan untuk mencari pembebanan optimal tiap kombinasi pada interval waktu tertentu di dalam program *unit commitment*. Tidak hanya digunakan pada *unit commitment*, *economic dispatch* juga digunakan untuk memeriksa hasil dari *unit commitment* apakah sudah memenuhi batasan *ramp-rate* atau belum.



Gambar 3. Flowchart iterasi lambda pada *economic dispatch*

IV. SIMULASI DAN ANALISIS

A. Studi Kasus 1

Studi kasus 1 memperlihatkan hasil perhitungan ketika data masukan menggunakan input linier. Program yang digunakan adalah program yang belum dikembangkan pada Tugas Akhir ini. Berikut ini adalah hasil dari simulasi *software*:

Tabel 1
Hasil simulasi studi kasus 1

Interval ke-	Unit status				Pcost (\$/hr)
	1	2	3	4	
1	OFF	ON	ON	OFF	9208
2	OFF	ON	ON	OFF	10648
3	OFF	ON	ON	ON	11974
4	OFF	ON	ON	OFF	10828
5	OFF	ON	ON	OFF	8308
6	OFF	ON	ON	OFF	6192
7	OFF	ON	ON	OFF	6366
8	OFF	ON	ON	OFF	10108

Total biaya pembangkitan sebesar 73633,40 \$

Dari simulasi di atas, terlihat bahwa unit pembangkit 2 dan 3 selalu terhubung dengan sistem (ON). Pada interval ke-3 dengan beban 600 MW, unit 2 dan 3 tidak mampu untuk memenuhi permintaan beban tersebut sehingga unit 4 diaktifkan.

Pada program *unit commitment* ini hanya menampilkan unit yang ON dan OFF saja, namun tidak dapat menampilkan berapa besar pembangkitan setiap pembangkitnya. Oleh karena itu pada studi kasus 2 dan 3 akan terlihat pembangkitan setiap unit pembangkit pada periode tertentu berdasarkan *software* yang telah dikembangkan

B. Studi Kasus 2

Pada studi kasus kedua ini memperlihatkan hasil perhitungan ketika data masukan menggunakan *input* nonlinier. Program yang digunakan adalah program yang telah dikembangkan untuk Tugas Akhir ini. Berikut ini adalah hasil dari simulasi *software*:

Tabel 2.
Hasil simulasi studi kasus 2

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	455	245	0	0	0	0
2	455	295	0	0	0	0
3	455	395	0	0	0	0
4	455	365	0	130	0	0
5	455	415	0	130	0	0
6	455	455	0	130	60	0
7	455	455	0	130	110	0
8	455	455	0	130	160	0
9	455	455	130	130	130	0
10	455	455	130	130	162	68
11	455	455	130	130	162	80
12	455	455	130	130	162	80
13	455	455	130	130	162	68
14	455	455	130	130	130	0
15	455	455	130	160	0	0
16	455	440	130	25	0	0
17	455	390	130	25	0	0
18	455	455	130	60	0	0
19	455	455	130	160	0	0
20	455	455	130	130	162	68
21	455	455	130	130	110	20
22	455	455	130	0	0	60
23	455	335	110	0	0	0
24	455	215	130	0	0	0

Tabel 2.
Hasil simulasi studi kasus 2 (lanjutan)

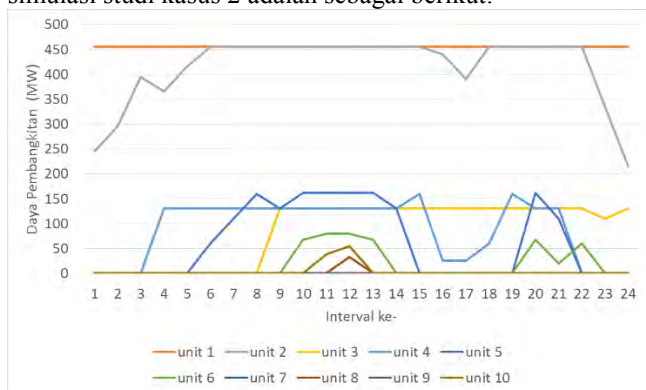
Interval ke-	Daya Pembangkitan (MW)				
	unit 7	unit 8	unit 9	unit 10	cost per interval (\$)
1	0	0	0	0	13665,9
2	0	0	0	0	14537,22
3	0	0	0	0	16283,77
4	0	0	0	0	18619,7
5	0	0	0	0	19494,36
6	0	0	0	0	21841,26
7	0	0	0	0	22859,35
8	0	0	0	0	23896,94

Tabel 2.
Hasil simulasi studi kasus 2 (lanjutan)

9	0	0	0	0	26163,69
10	0	0	0	0	28754,03
11	0	0	0	38	30686
12	0	33	0	55	32700
13	0	0	0	0	28754
14	0	0	0	0	26164
15	0	0	0	0	23897
16	0	0	0	0	20877,23
17	0	0	0	0	20001,75
18	0	0	0	0	21841,26
19	0	0	0	0	23896,94
20	0	0	0	0	28754,03
21	0	0	0	0	26571,31
22	0	0	0	0	21963,64
23	0	0	0	0	17785,39
24	0	0	0	0	16035,83

Dengan total biaya pembangkitan 548996,64 \$

Jika direpresentasikan dalam bentuk grafik, maka hasil simulasi studi kasus 2 adalah sebagai berikut:



Gambar 3. Grafik pembebanan studi kasus 2

Berdasarkan gambar 3, terlihat bahwa unit 1 selalu *commit (on)* dengan pembangkitan maksimal. Unit 2 menjadi pembangkit kedua yang pembangkitannya terbesar. Hal ini karena unit 1 dan unit 2 memiliki fungsi biaya termurah diantara pembangkit yang lain.

Pada studi kasus 2 ini efek *ramp-rate* tidak terlalu terlihat. Meskipun pada data *input* terdapat UR dan DR sebagai batasan *ramp-rate*, namun UR dan DR tersebut memiliki nilai yang besar. Besarnya batasan *ramp-rate* tersebut menyebabkan perubahan besar pembangkitan pada unit yang aktif tidak selalu masuk dalam batasan *ramp-rate*.

Efek dari batasan *minimum up-time* dan *minimum down-time* juga berpengaruh pada hasil simulasi studi kasus 2. Unit 3 mulai *commit* saat memasuki interval ketiga karena batasan *minimum down-time* sudah terpenuhi saat interval ketiga.

C. Studi Kasus 3

Pada studi kasus ketiga ini memperlihatkan hasil perhitungan ketika data masukan menggunakan *input* nonlinier dengan memperhatikan batasan *ramp-rate*. Program yang digunakan adalah program yang telah dikembangkan untuk Tugas Akhir ini.

Data masukan yang digunakan adalah sistem 10 unit pembangkit dengan 8 interval pembebanan. Pada data masukan minimum *up time*, minimum *down time*, start up cost, dan initial condition diabaikan atau dianggap nol. Namun efek *ramp-rate* diperhitungkan dengan nilai UR dan DR yang relatif kecil, tidak seperti studi kasus 2 yang UR dan DR memiliki nilai yang besar. Berikut ini adalah hasil dari simulasi *software*:

Tabel 3.
Hasil simulasi studi kasus 3

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	163,20
4	0	0	0	70	93	163,20
5	0	90,52	0	70	93	216,17
6	0	93	0	70	93	253,01
7	53,85	93	108,88	70	93	240,25
8	73	93	134,43	70	93	278,58
Interval ke-	Daya Pembangkitan (MW)					cost per interval (\$)
	unit 7	unit 8	unit 9	unit 10		
1	100	190	190	0		947
2	147,99	190	190	113		1286
3	180,80	190	190	113		1961
4	180,80	190	190	113		2130
5	228,30	190	190	113		2610
6	248	190	190	113		2769
7	248	190	190	113		3175
8	248	190	190	113		3412
Total biaya pembangkitan						18290

Berdasarkan hasil simulasi, beberapa unit pembangkit melanggar batasan *ramp-rate*. Tabel 4. Merepresentasikan interval dan pembangkit yang melanggar *ramp-rate*.

Tabel 4.
Interval yang masih melanggar *ramp-rate*

Unit	interval	Kenaikan/ Penurunan daya (MW)	UR/DR
1	7 ke 8	+19.14518	10
3	7 ke 8	+25.54273	23
6	4 ke 5	+52.97414	50

Oleh karena pada hasil *unit commitment* terdapat unit pembangkit yang belum memenuhi *ramp-rate*, maka diperlukan perhitungan kembali dengan program *economic dispatch* dengan memperhatikan *ramp-rate*.

Dengan data masukan yang sama, program *economic dispatch* digunakan untuk mencari pembebanan optimal dengan memperhitungkan *ramp-rate*. Data dihitung setiap interval. Sedangkan *ramp-rate* diasumsikan digunakan pada setiap perubahan pembangkitan unit pembangkit yang ON.

Untuk perubahan dari pembangkit yang OFF menjadi ON diasumsikan tidak menggunakan batasan *ramp-rate*.

Berikut ini adalah hasil simulasi setelah dikoreksi program *economic dispatch* dengan memperhatikan *ramp-rate*:

Tabel 5.

Hasil simulasi studi kasus 3 dengan *ramp-rate*

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	156,7
4	0	0	0	70	93	156,7
5	0	93	0	70	93	206,7
6	0	93	0	70	93	253
7	53,7	93	115,4	70	93	233,9
8	63,7	93	1138,4	70	93	283,9

Tabel 5.

Hasil simulasi studi kasus 3 dengan *ramp-rate* (lanjutan)

Interval ke-	Daya Pembangkitan (MW)				cost per interval (\$)
	unit 7	unit 8	unit 9	unit 10	
1	100	190	190	0	947
2	148	190	190	113	1290,71
3	187,3	190	190	113	1969,75
4	187,3	190	190	113	2139,13
5	235,3	190	190	113	2619,9
6	248	190	190	113	2780,17
7	248	190	190	113	3180,32
8	248	190	190	113	3417,8
Total biaya pembangkitan					18344,78

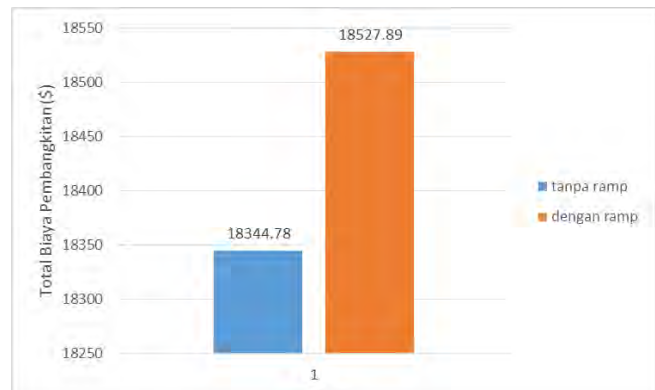
Berdasarkan hasil pada Tabel 5. pembangkit sudah memenuhi *ramp-rate*. Tabel 6. merepresentasikan *ramp-rate* yang sudah terpenuhi.

Tabel 6.

Interval yang sudah tidak melanggar *ramp-rate*

Unit	interval	Kenaikan/ Penurunan daya (MW)	UR/DR
1	7 ke 8	+10	10
3	7 ke 8	+23	23
6	4 ke 5	+50	50

Jika hasil simulasi yang pertama dan kedua dibandingkan, maka hasil simulasi yang kedua memiliki total biaya pembangkitan yang lebih tinggi. Hal ini karena hasil simulasi yang kedua memperhatikan batasan *ramp-rate*, sehingga perhitungan pembangkitan pada setiap semakin ketat dan kompleks.



Gambar 4. Total biaya perhitungan dengan *ramp-rate* dan tidak dengan *ramp-rate*

Dari Gambar 4 terlihat bahwa ketika *ramp-rate* diperhitungkan akan memiliki biaya yang lebih tinggi daripada perhitungan yang mengabaikan *ramp-rate*. Perbedaan harganya adalah sebesar 183,11 \$. Hal ini dikarenakan batasan *ramp-rate* menyebabkan variasi daya pembangkitan yang lebih ketat.

V. KESIMPULAN

Dari hasil simulasi dan analisis diperoleh kesimpulan sebagai berikut:

1. Aplikasi perhitungan *unit commitment* dapat melakukan perhitungan sesuai batasan minimum up time dan minimum down time dengan dengan *input* nonlinier. Pada studi kasus 2 hasil output perhitungan didapatkan total biaya pembangkitan 548.996,64 \$.
2. Diperlukan bantuan aplikasi perhitungan *economic dispatch* jika hasil keluaran dari *unit commitment* melanggar *ramp-rate*. Aplikasi perhitungan *economic dispatch* sudah dapat menghitung pembebanan optimal pada setiap pembangkit dengan mempertimbangkan *ramp-rate*.
3. Total biaya pembangkitan dengan mempertimbangkan *ramp-rate* lebih besar daripada tanpa batasan *ramp-rate* karena batasan *ramp-rate* menyebabkan variasi daya pembangkitan yang lebih ketat. Pada hasil studi kasus 3, total biaya pembangkitan dengan mempertimbangkan *ramp-rate* lebih mahal 183,11 \$.

DAFTAR PUSTAKA

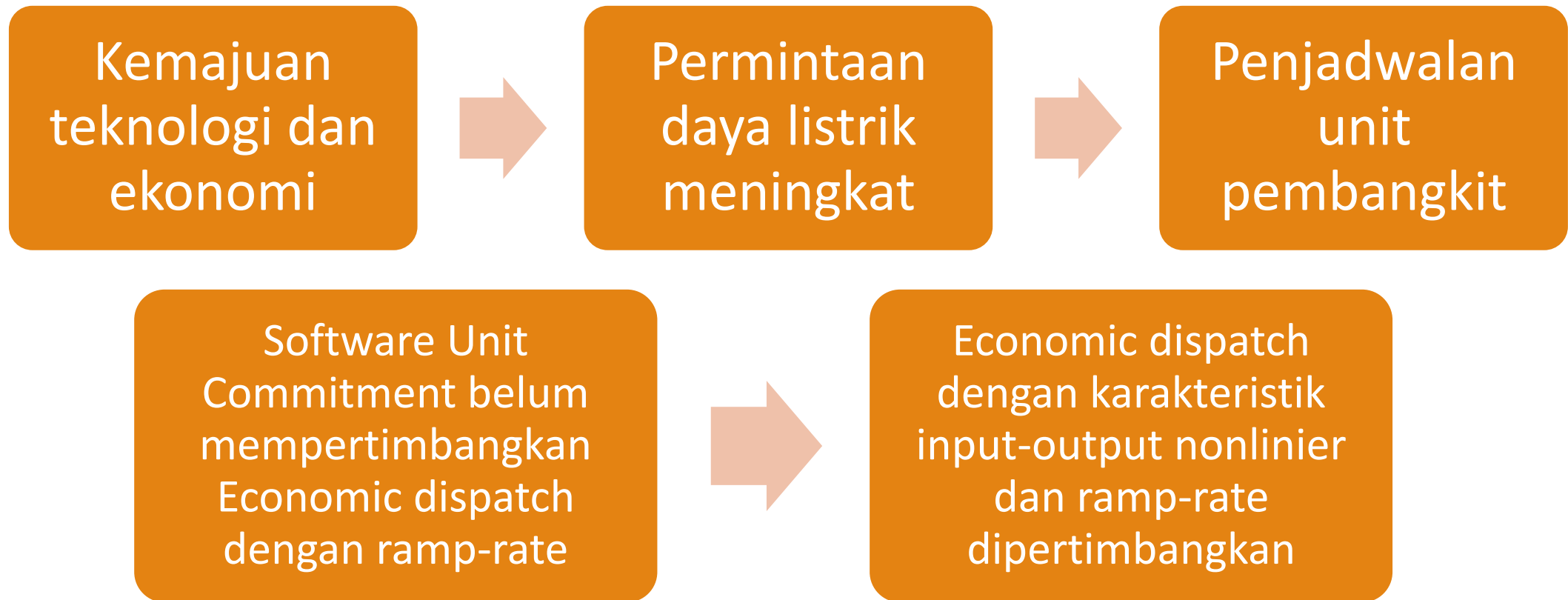
- [1] Wood, A.J., dan Bruce F.W., "Power Generation, Operation, and Control", A Wiley-Interscience Publication. New York, 1984.
- [2] Saadat, Hadi. "Power System Analysis 2nd Edition", McGrawHill, Ch.1, 1999.
- [3] Wang, C., dan Shahidehpour, S. M., "Effect of Ramp-Rate Limits on Unit Commitment and Economic Dispatch", IEEE Paper, 1993.
- [4] Sundaram, C. S., Sudhakaran, M., Selvakumar, R., Natarajan, R. G., "Solution to Profit Based Unit Commitment using Swarm Intelligence Technique", Proc. of ICCPEIC, 2014.
- [5] Abidin, Zainal, "Dynamic Economic Dispatch Menggunakan Quadratic Programming", Buku Tugas Akhir, Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2012.

Pengembangan *Software Unit* *Commitment And Economic Dispatch* dengan Mempertimbangkan *Ramp-Rate*

Wildan Nurwanto---2211100167

Dosen Pembimbing
Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D.
Ir. Arif Musthofa, MT.

Latar Belakang



Permasalahan

1. Bagaimana membuat program *unit commitment* dengan input nonlinier berbasis Delphi?
2. Bagaimana melakukan *economic dispatch* dengan metode iterasi lambda pada *unit commitment*?
3. Bagaimana melakukan pengecekan hasil *unit commitment* agar tetap memenuhi batasan *ramp-rate*?

Batasan Masalah

1. Pembuatan program menggunakan Delphi.
2. Rugi-rugi diabaikan.
3. Metode yang digunakan untuk menghitung *economic dispatch* dengan input non linier adalah iterasi lambda.
4. Metode yang digunakan untuk menghitung *unit commitment* adalah *dynamic programming*.
5. Data beban tidak melebihi kemampuan pembangkitan total pembangkit yang aktif.
6. Sistem dalam keadaan stabil.
7. Data input hanya menggunakan satu jenis *start up cost*.
8. Tidak mempertimbangkan biaya padam.

Tujuan

1. Membuat *software unit commitment* dengan input *nonlinier* orde dua.
2. Membuat *software unit commitment and economic dispatch* dengan memperhatikan *ramp-rate*.
3. Melihat pengaruh ramp-rate pada perhitungan *unit commitment and economic dispatch*.

Fungsi objektif

$$F_{cost}(K, I) = MIN[P_{cost}(K, I) + S_{cost}(K - 1, L, K, I)]$$

Equality constraint

$$\sum_{i=1}^T P_i(t) = P_{load}(t)$$

Inequality constraint

$$P_i^{min} \leq P_i(t) \leq P_i^{max}$$

Min up and down time

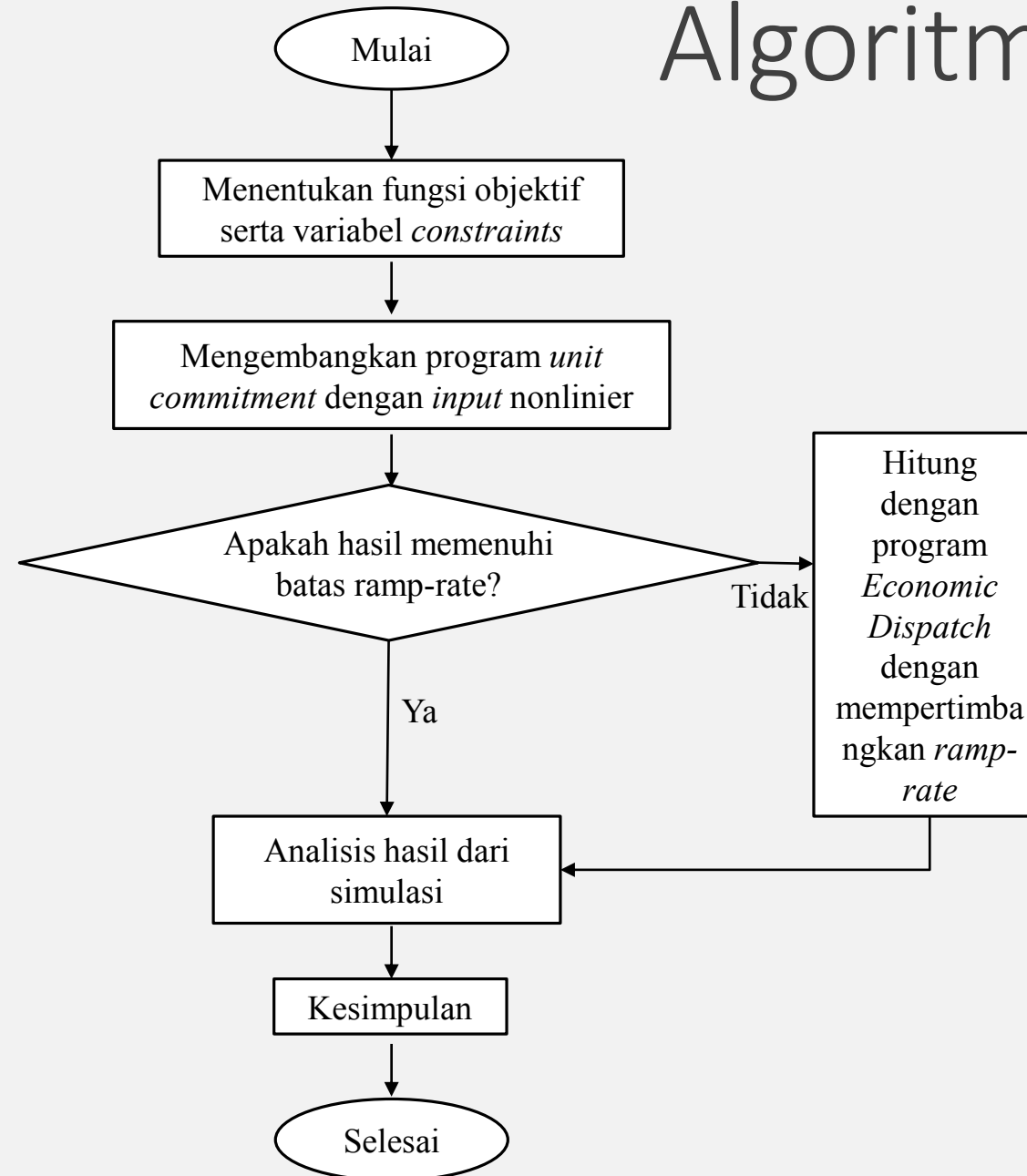
$$[X_i^{on}(t-1) - T_i^{on}] * [I_i(t-1) - I_i] \geq 0$$

$$[X_i^{off}(t-1) - T_i^{off}] * [I_i - I_i(t-1)] \geq 0$$

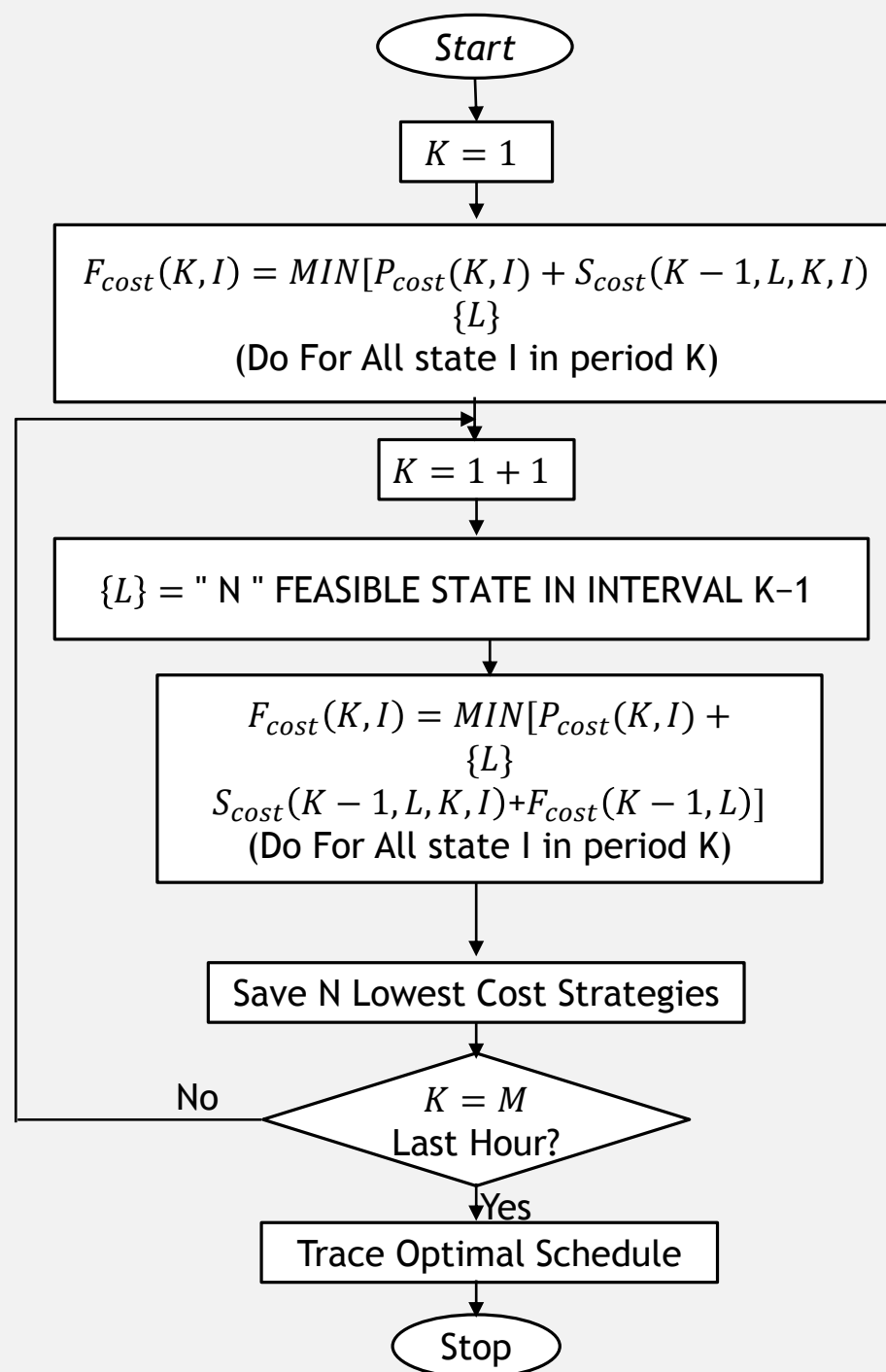
Ramp-rate

$$P_i(t-1) - DR_i \leq P_i(t) \leq UR_i + P_i(t-1)$$

Algoritma Pengerjaan



Unit commitment



Unit commitment

Cari kombinasi yang mungkin = lihat maksimal pembangkitan dgn beban.

Tiap kombinasi yang mungkin dicari pembebanan optimalnya dengan economic dispatch.

Kemudian bandingkan kombinasi-kombinasi yang ada, ambil yang paling murah. Didapatlah hasil unit commitment untuk interval 1.

Untuk interval 2, langkahnya sama seperti di atas, namun perlu diperhatikan start-up cost, minimum up-time, minimum down-time.

Setelah, mendapatkan penjadwalan unit pembangkit dan pembangkitannya, kemudian dicek ramp-ratenya.

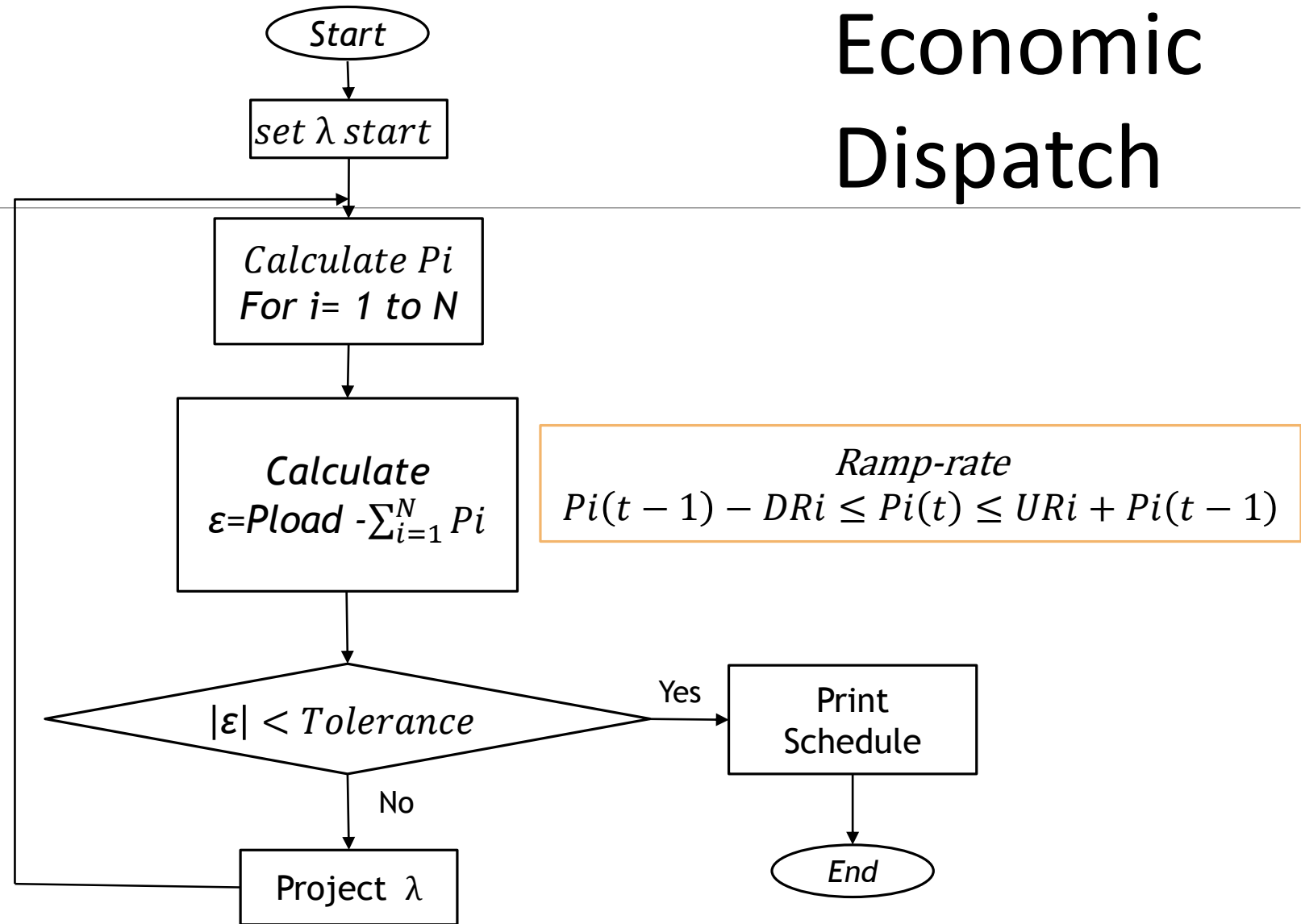
Jika melanggar ramp-rate, hasil dihitung kembali dengan program economic dispatch.

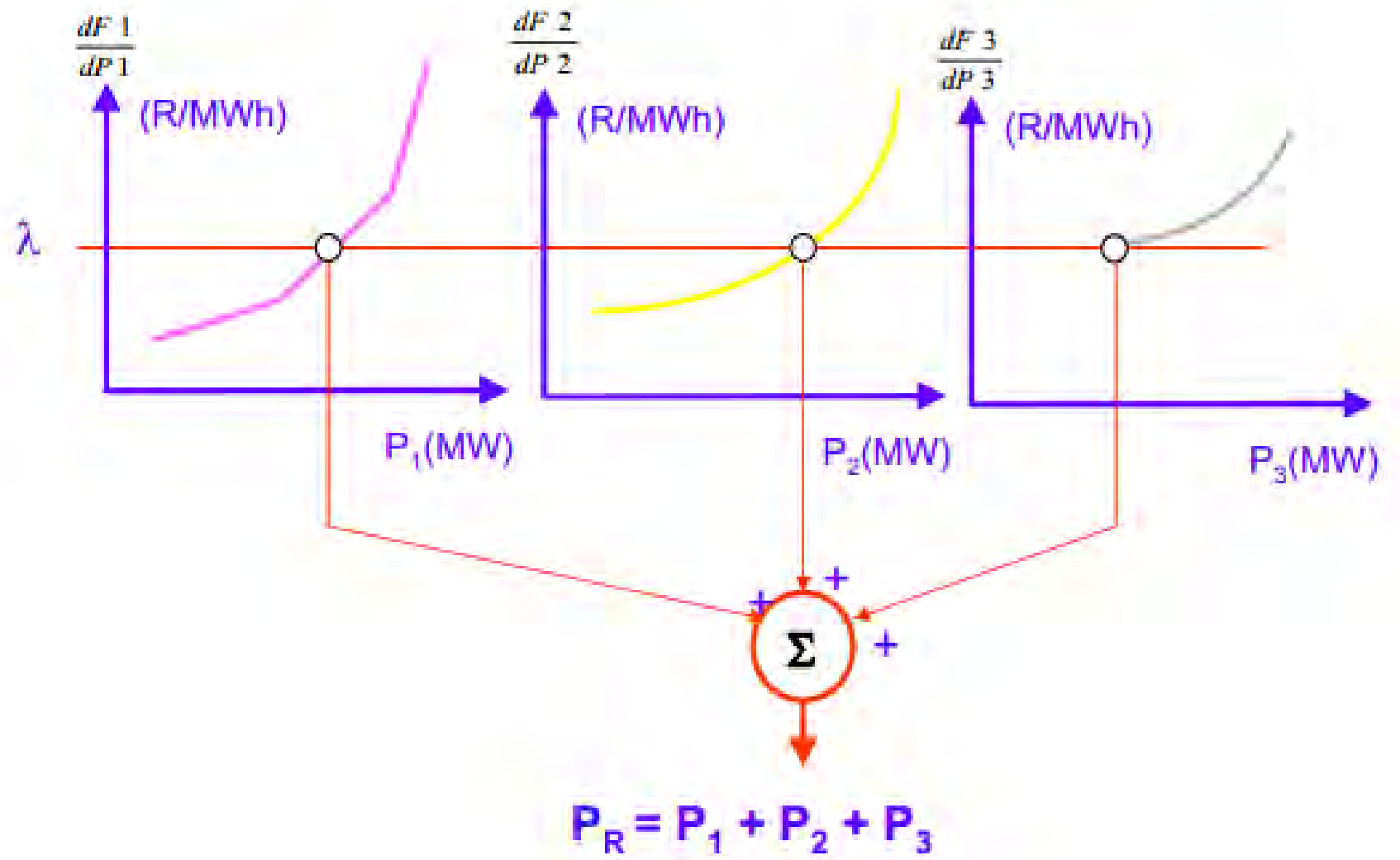
Economic Dispatch

$$\frac{\partial F_i}{\partial P_i} = \lambda \left(1 - \frac{\partial P_{loss}}{\partial P_i} \right)$$
$$\lambda = \frac{\partial F_i}{\partial P_i} \left(\frac{1}{1 - \frac{\partial P_{loss}}{\partial P_i}} \right)$$
$$\lambda = \frac{\partial F_i}{\partial P_i} P_{fi}$$

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2}$$

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2}$$





$$P_i = \frac{\lambda - b}{2a}$$

Studi Kasus I

Unit Commitment
Dengan input Linier
(EDC)

Studi Kasus II

Unit Commitment
Dengan input Non-
Linier(EDC)

Studi Kasus III

Unit Commitment input
Non-Linier(EDC)
Kemudian dicek sudah
memenuhi ramp atau
belum

Studi Kasus 1

(Unit Commitment Dengan input Linier (EDC))

Data input

unit	Max (MW)	Min (MW)	Incremental Heat Rate (Btu/KHh)	No Load Cost (R/h)	Start Up Cost (R/mW h)	Fuel Cost
1	25	80	10440	213	350	2
2	60	250	9000	585.6	400	2
3	75	300	8730	684.7	1100	2
4	20	60	11900	252	0	1.2

Interval ke-	Beban (MW)
1	450
2	530
3	600
4	540
5	400
6	280
7	290
8	500

Studi Kasus 1

(Unit Commitment Dengan input Linier (EDC))

unit	Min Up Time	Min down time	Initial status
1	4	2	-5
2	5	3	8
3	5	4	8
4	1	1	-6

Studi Kasus 1

(Unit Commitment dengan input Linier (EDC))

Hasil:

Interval ke-	Unit status				Pcost (\$/hr)
	1	2	3	4	
1	OFF	ON	ON	OFF	9208
2	OFF	ON	ON	OFF	10648
3	OFF	ON	ON	ON	11974
4	OFF	ON	ON	OFF	10828
5	OFF	ON	ON	OFF	8308
6	OFF	ON	ON	OFF	6192
7	OFF	ON	ON	OFF	6366
8	OFF	ON	ON	OFF	10108

Total biaya pembangkitan 73633.40 \$

Studi Kasus 2

(Unit Commitment dengan input nonlinier(EDC))

Data input

Keterangan	Unit Pembangkit					Keterangan	Unit Pembangkit				
	1	2	3	4	5		6	7	8	9	10
Pmax	455	455	130	130	162	Pmax	80	85	55	55	55
Pmin	150	150	20	20	25	Pmin	20	25	10	10	10
a	0.000	0.000	0.002	0.002	0.003	a	0.007	0.007	0.002	0.001	0.004
	4	3		1	9		1	9	2	7	1
b	16.19	17.26	16.6	16.5	19.7	b	22.36	27.74	27.27	27.79	25.92
c	1000	970	700	680	450	c	370	480	665	670	660
MUT	8	8	5	5	6	MUT	3	3	1	1	1
MDT	8	8	5	5	6	MDT	3	3	1	1	1
UR	100	100	130	130	162	UR	80	85	55	55	55
DR	120	120	130	130	162	DR	80	85	55	55	55
Start up cost	4500	5000	550	560	900	Start up cost	170	260	30	30	30
Initial status	8	8	-5	-5	-6	Initial status	-3	-3	-1	-1	-1

Studi Kasus 2

(*Unit Commitment* dengan input nonlinier(EDC))

Data input

Interval ke-	Beban (MW)	Interval ke-	Beban (MW)
1	700	13	1400
2	750	14	1300
3	850	15	1200
4	950	16	1050
5	1000	17	1000
6	1100	18	1100
7	1150	19	1200
8	1200	20	1400
9	1300	21	1300
10	1400	22	1100
11	1450	23	900
12	1500	24	800

mempertimbangkan:

- Min up and down time
- Start up cost

Studi Kasus 2

Hasil

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	455	245	0	0	0	0
2	455	295	0	0	0	0
3	455	395	0	0	0	0
4	455	365	0	130	0	0
5	455	415	0	130	0	0
6	455	455	0	130	60	0
7	455	455	0	130	110	0
8	455	455	0	130	160	0
9	455	455	130	130	130	0
10	455	455	130	130	162	68
11	455	455	130	130	162	80
12	455	455	130	130	162	80
13	455	455	130	130	162	68
14	455	455	130	130	130	0
15	455	455	130	160	0	0
16	455	440	130	25	0	0
17	455	390	130	25	0	0
18	455	455	130	60	0	0
19	455	455	130	160	0	0
20	455	455	130	130	162	68
21	455	455	130	130	110	20
22	455	455	130	0	0	60
23	455	335	110	0	0	0
24	455	215	130	0	0	0

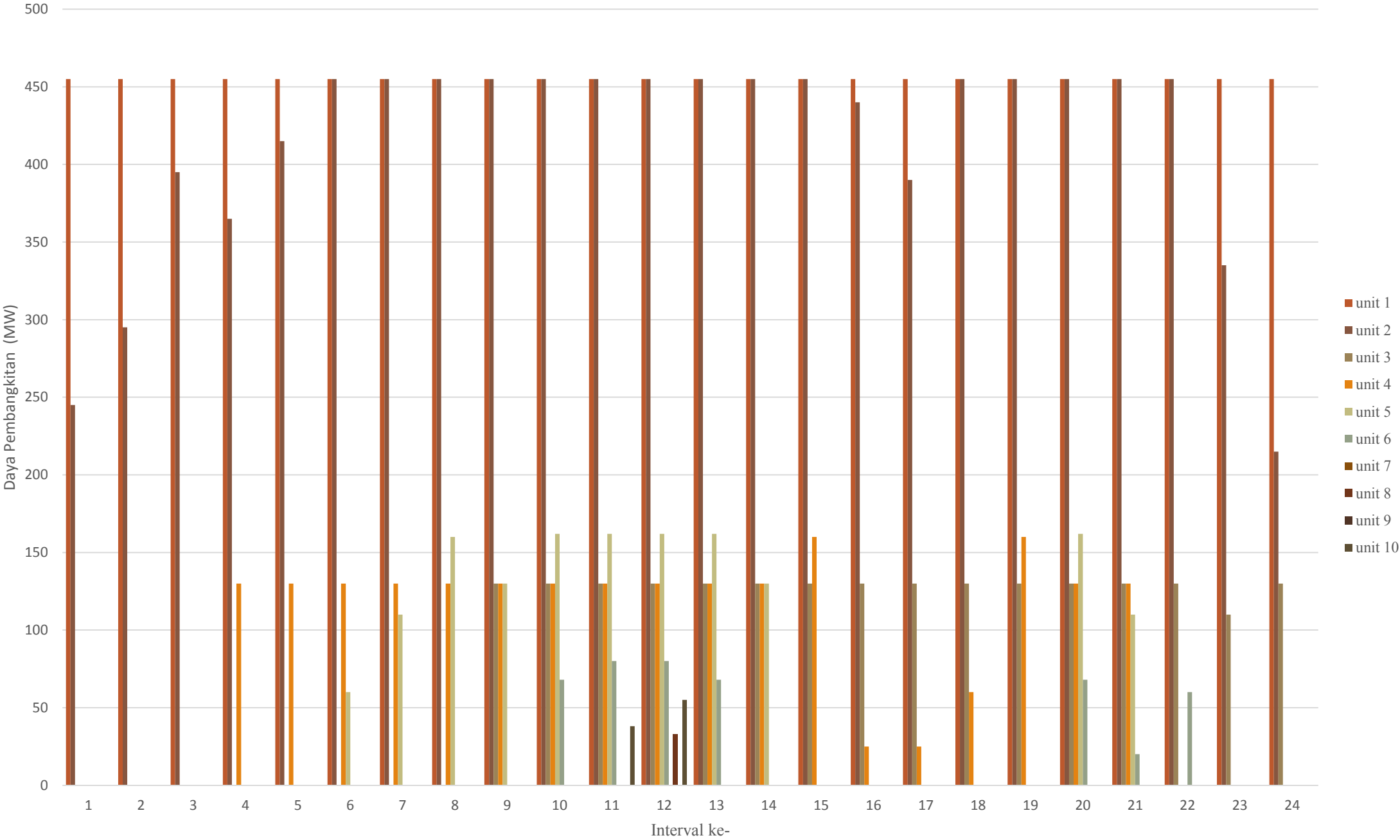
Studi Kasus 2

Hasil

Total biaya pembangkitan 548996.64 \$

Interval ke-	Daya Pembangkitan (MW)					cost per interval (\$)
	unit 7	unit 8	unit 9	unit 10		
1	0	0	0	0		13665,9
2	0	0	0	0		14537,22
3	0	0	0	0		16283,77
4	0	0	0	0		18619,7
5	0	0	0	0		19494,36
6	0	0	0	0		21841,26
7	0	0	0	0		22859,35
8	0	0	0	0		23896,94
9	0	0	0	0		26163,69
10	0	0	0	0		28754,03
11	0	0	0	38		30686
12	0	33	0	55		32700
13	0	0	0	0		28754
14	0	0	0	0		26164
15	0	0	0	0		23897
16	0	0	0	0		20877,23
17	0	0	0	0		20001,75
18	0	0	0	0		21841,26
19	0	0	0	0		23896,94
20	0	0	0	0		28754,03
21	0	0	0	0		26571,31
22	0	0	0	0		21963,64
23	0	0	0	0		17785,39
24	0	0	0	0		16035,83

Grafik Pembangkitan Studi Kasus 2



Studi Kasus 3

(Unit Commitment input nonlinier(EDC) kemudian dicek sudah memenuhi ramp atau belum)

Data Input

Tabel 4.11 Data input studi kasus 3

Keterangan	Unit Pembangkit				
	1	2	3	4	5
Pmax (MW)	73	93	143	70	93
Pmin (MW)	12	26	42	18	30
a	0.0051	0.0040	0.0039	0.0038	0.0021
b	2.2034	1.9101	1.8518	1.6966	1.8015
c	15	25	40	32	29
UR	10	15	23	10	13
DR	10	15	23	10	13

Tabel 4.11 Data input studi kasus 3 (lanjutan)

Keterangan	Unit Pembangkit				
	6	7	8	9	10
Pmax	350	248	190	190	113
Pmin	100	100	40	70	40
a	0.0026	0.0029	0.0015	0.0013	0.0014
b	1.5354	1.2643	1.2130	1.1954	1.1285

Tabel 4.12 Data input studi kasus 3 (lanjutan)

c	72	49	82	105	100
UR	50	48	20	30	20
DR	50	48	20	30	20

Interval ke-	Beban (MW)
1	480
2	641
3	930
4	1000
5	1191
6	1250
7	1400
8	1483

Studi Kasus 3

Hasil

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	163.20
4	0	0	0	70	93	163.20
5	0	90.52	0	70	93	216.17
6	0	93	0	70	93	253.01
7	53.85	93	108.88	70	93	240.25
8	73	93	134.43	70	93	278.58

Interval ke-	Daya Pembangkitan (MW)				
	unit 7	unit 8	unit 9	unit 10	cost per interval (\$)
1	100	190	190	0	947
2	147.99	190	190	113	1286
3	180.80	190	190	113	1961
4	180.80	190	190	113	2130
5	228.30	190	190	113	2610
6	248	190	190	113	2769
7	248	190	190	113	3175
8	248	190	190	113	3412
Total biaya pembangkitan					18290

Studi Kasus 3

ramp

Unit	interval	Kenaikan/ Penurunan daya (MW)	UR/DR
1	7 ke 8	+19.14518	10
3	7 ke 8	+25.54273	23
6	4 ke 5	+52.97414	50

Studi Kasus 3

Hasil dengan ramp

Interval ke-	Daya Pembangkitan (MW)					
	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	93	156.7
4	0	0	0	70	93	156.7
5	0	93	0	70	93	206.7
6	0	93	0	70	93	253
7	53.7	93	115.4	70	93	233.9
8	63.7	93	1138.4	70	93	283.9

Interval ke-	Daya Pembangkitan (MW)				
	unit 7	unit 8	unit 9	unit 10	cost per interval (\$)
1	100	190	190	0	947
2	148	190	190	113	1290.71
3	187.3	190	190	113	1969.75
4	187.3	190	190	113	2139.13
5	235.3	190	190	113	2619.9
6	248	190	190	113	2780.17
7	248	190	190	113	3180.32
8	248	190	190	113	3417.8
Total biaya pembangkitan					18344.78

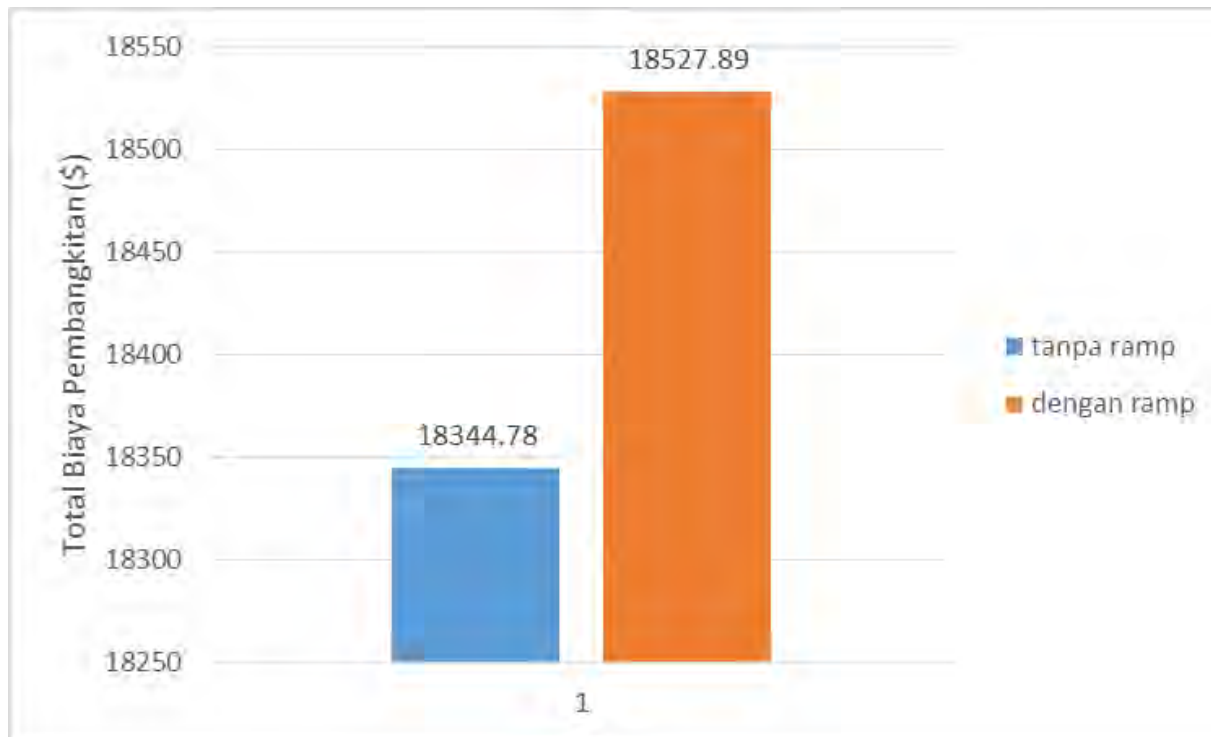
Studi Kasus 3

ramp

Unit	interval	Kenaikan/ (MW)	Penurunan daya	UR/DR
1	7 ke 8		+10	10
3	7 ke 8		+23	23
6	4 ke 5		+50	50

Studi Kasus 3

Perbandingan total biaya



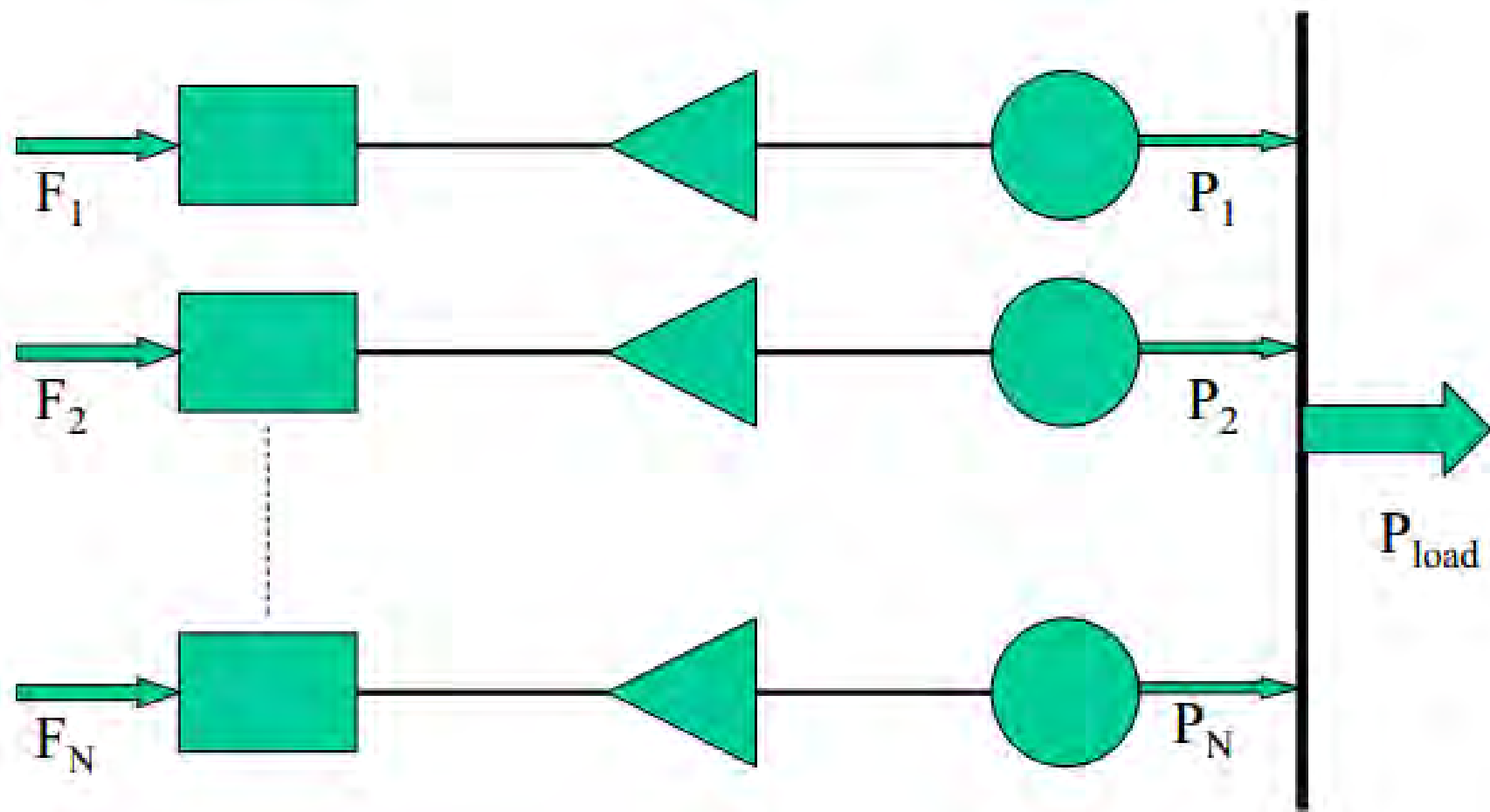
Kesimpulan

1. Aplikasi perhitungan unit commitment dapat melakukan perhitungan sesuai batasan minimum up time dan minimum down time dengan dengan input nonlinier.
2. Diperlukan bantuan aplikasi perhitungan economic dispatch jika hasil keluaran dari unit commitment melanggar ramp-rate. Aplikasi perhitungan economic dispatch sudah dapat menghitung pembebanan optimal pada setiap pembangkit dengan mempertimbangkan ramp-rate.
3. Total biaya pembangkitan dengan mempertimbangkan ramp-rate lebih besar daripada tanpa batasan ramp-rate karena batasan ramp-rate menyebabkan variasi daya pembangkitan yang lebih ketat.

Saran

1. Software ini dapat ditambahkan spinning reserve.
2. Permasalahan dapat dikembangkan dengan menambahkan rugi saluran transmisi.

Terimakasih



Problem Formulation :

The objective function, F_T , is equally to the total cost for supplying the indicated load.

The problem is to **minimize** F_T subject to the constraint that the sum of the powers generated must equal the received load.

Transmission losses are neglected

$$F_T = F_1 + F_2 + F_3 + \dots + F_N$$

$$= \sum_{i=1}^N F_i(P_i)$$

$$\phi = 0 = P_{load} - \sum_{i=1}^N P_i$$

LAGRANGE function :

$$\mathcal{L} = F_T + \lambda \phi$$

The necessary conditions for an **extreme value** of the objective function :

$$\frac{\partial \mathcal{L}}{\partial P_i} = \frac{dF_i(P_i)}{dP_i} - \lambda = 0 \quad , \text{ atau}$$

$$0 = \frac{dF_i}{dP_i} - \lambda$$

Validasi program

No. Unit	System Input	Input-Output Curve	Output-Max (MW)	Input-Min (MW)	Fuel Cost (R/Mbtu)
Unit 1	Coal-Fired Steam	$H_1\left(\frac{MBtu}{h}\right) = 510.0 + 7.2P_1 + 0.00142P_1^2$	600	150	1.1
Unit 2	Oil-Fired Steam	$H_2\left(\frac{MBtu}{h}\right) = 310.0 + 7.85P_2 + 0.00194P_2^2$	400	100	1.0
Unit 3	Oil-Fired Steam	$H_3\left(\frac{MBtu}{h}\right) = 78.0 + 7.97P_3 + 0.00482P_3^2$	200	50	1.0

Karakteristik Input-Output dalam R/h :

$$F_1(P_1) = H_1(P_1) \times 1.1 = 561 + 7.92 P_1 + 0.001562P_1^2 \text{ R/h}$$

$$F_2(P_2) = H_2(P_2) \times 1.0 = 310 + 7.85 P_2 + 0.00194P_2^2 \text{ R/h}$$

$$F_3(P_3) = H_3(P_3) \times 1.0 = 78 + 7.97 P_3 + 0.00482P_3^2 \text{ R/h}$$

hasil

generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	393.2		9.1483	1.0000	3916.38
UNIT2	334.6		9.1483	1.0000	3153.86
UNIT3	122.2		9.1483	1.0000	1124.16
totals	850.0				8194.40
lambda =	9.1483				

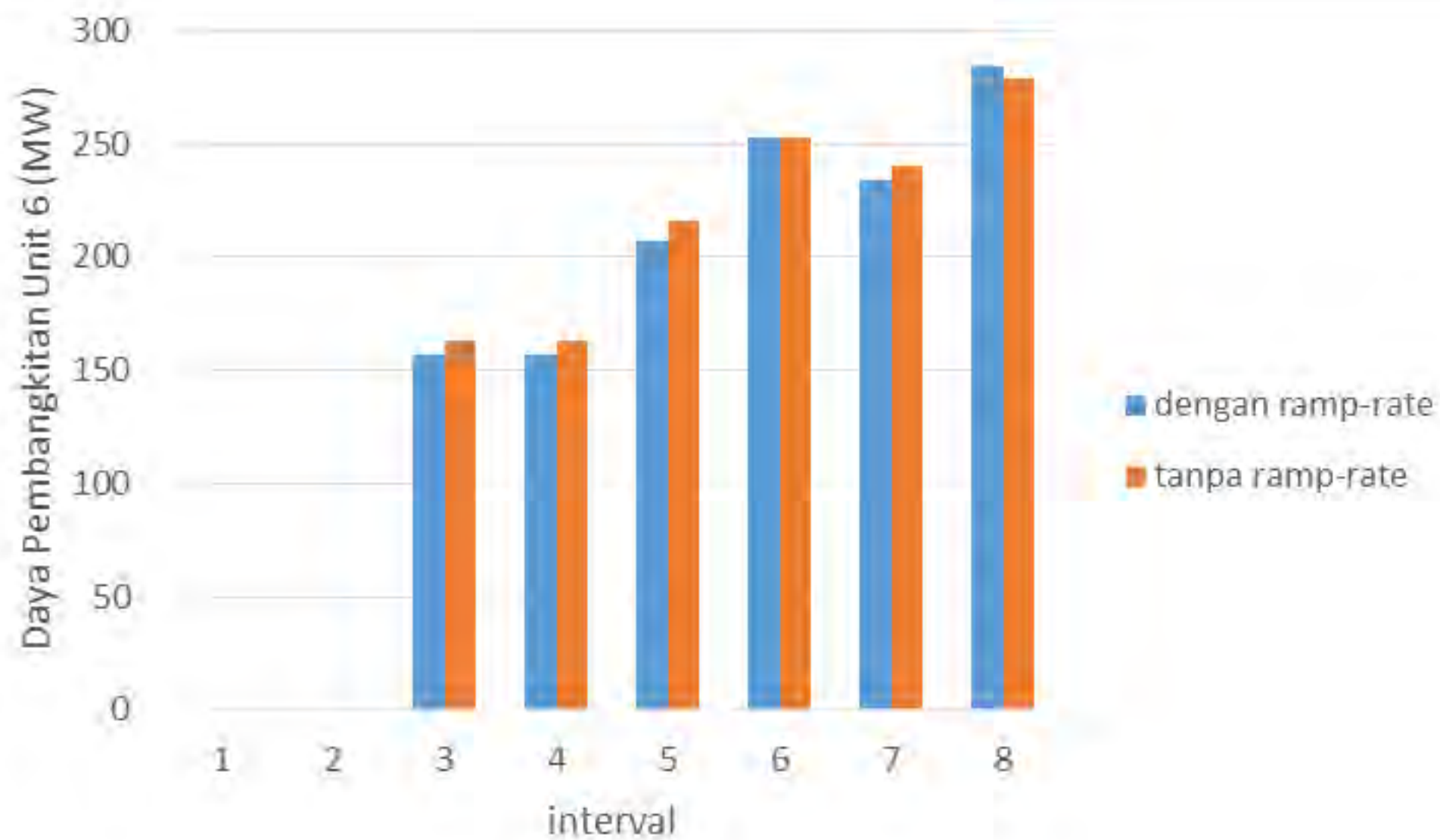
OPTIMUM COMMITMENT SCHEDULE

TOTAL COST = 8194.40 R

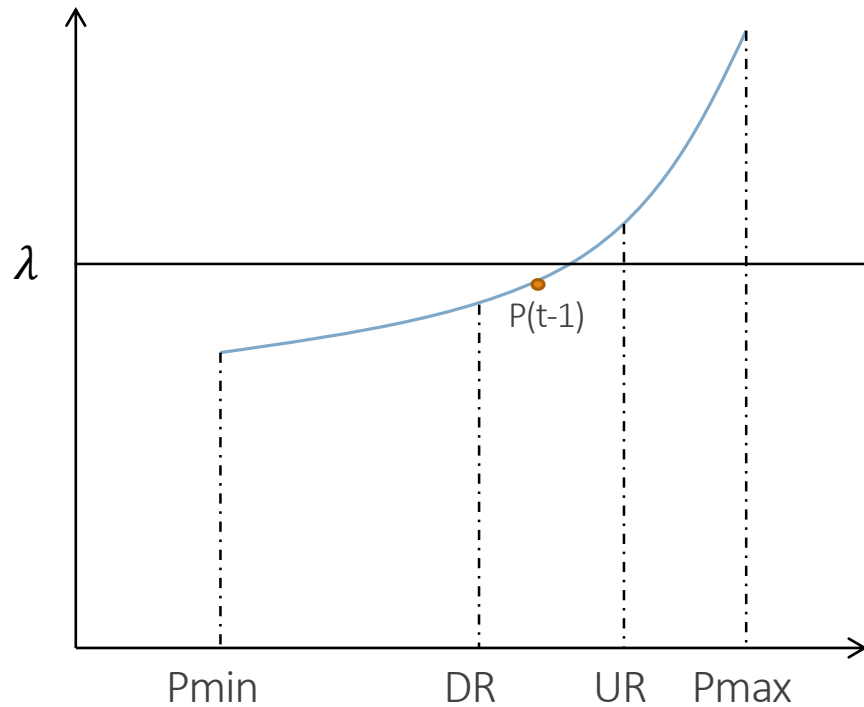
PERIOD	STATE	UNIT STATUS			PCOST R/HR	LOAD MW
		1	2	3		
1	1	ON	ON	ON	8194	850.000

Hasil UC

Bangkit = 393.17205
Bangkit = 334.60553
Bangkit = 122.22712



Ramp rate



Batasan ramp-rate
menyebabkan variasi
daya pembangkitan
lebih ketat

STAGE
AWAL

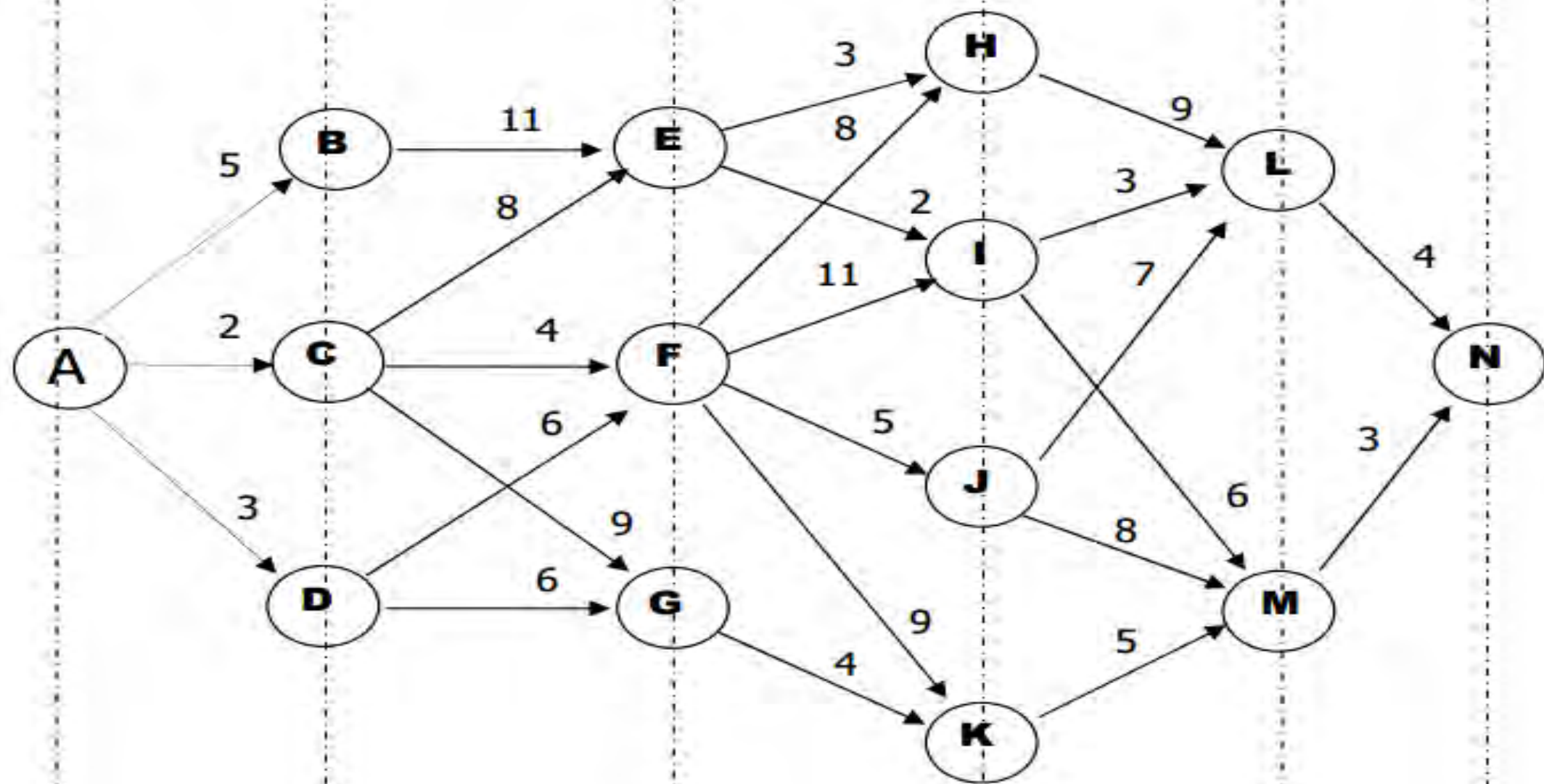
STAGE 1

STAGE 2

STAGE 3

STAGE 4

STAGE 5



Hasil

